

UNIVERSIDAD CARLOS III DE MADRID

ESCUELA POLITÉCNICA SUPERIOR



INGENIERÍA EN INFORMÁTICA

PROYECTO FIN DE CARRERA

**SISTEMA DE ATENCIÓN VISUAL PARA LA
DETECCIÓN DE PUNTOS TOPOLÓGICOS DE
REFERENCIA**

Autor: Andrés Duque Fernández

Tutor: Raúl Arrabales Moreno

Septiembre 2009

Proyecto Fin de Carrera.
Sistema de atención visual para la detección de puntos topológicos de referencia.

Agradecimientos:

*A mis padres y mi hermano,
por aguantarme en momentos de tensión
y felicitarme en momentos de éxito.*

A mis amigos, por todo lo que vivimos día a día.

*A mi tutor, por su amabilidad y disponibilidad
a lo largo del desarrollo de este proyecto.*

*Y por supuesto, a los Cebollitas, porque sin ellos
estos años de carrera no habrían sido lo mismo.*

Proyecto Fin de Carrera.
Sistema de atención visual para la detección de puntos topológicos de referencia.

TABLA DE CONTENIDOS

1. Introducción	9
1.1. Escenario del proyecto	9
1.2. Objetivos.....	10
1.3. Gestión del proyecto.....	11
1.4. Estructura de la memoria.....	13
2. Estado del arte	14
2.1. Robótica cognitiva.....	14
2.2. Robots autónomos móviles	16
2.2.1. Locomoción	17
2.2.2. Sensores	19
2.2.3. Localización de robots móviles	20
2.2.4. Planificación y navegación.....	21
2.3. Visión artificial.....	25
2.4. Atención visual	29
2.4.1. Funciones de la atención.....	31
2.4.2. Modelos psicológicos y neurológicos de la atención	32
2.4.3. Modelos computacionales de atención visual	34
2.5. Marcado topológico del terreno (landmarking).....	36
2.5.1. Reconocimiento de patrones.....	37
2.5.2. Detección de cuadriláteros por agrupamiento de aristas	39
2.5.3. Marcas de terreno atencionales.....	40
2.5.4. Detección de marcas basada en operadores de color.....	43
3. Herramientas Utilizadas	45
3.1. Microsoft Robotics Developer Studio.....	45
3.2. Lenguaje de programación (C#)	51
3.3. Navegador Web	51
3.4. Microsoft Office.....	51
3.5. Pioneer 3-DX.....	51
4. Objetivos.....	53

Proyecto Fin de Carrera.
Sistema de atención visual para la detección de puntos topológicos de referencia.

5.	Descripción del trabajo realizado	56
5.1.	<i>Requisitos de la aplicación.....</i>	57
5.2.	<i>Arquitectura de la aplicación.....</i>	64
5.3.	<i>Funcionamiento de la aplicación</i>	68
5.3.1.	Fase de detección.....	68
5.3.2.	Fase de localización.....	78
5.4.	<i>Descripción de los módulos y clases</i>	84
5.4.1.	Servicio <i>Landmarker</i>	86
5.4.2.	Servicio de control.....	90
5.4.3.	Ficheros externos	93
5.5.	<i>Manual de usuario.....</i>	94
6.	Experimentación y resultados	97
6.1.	<i>Descripción del entorno de pruebas.....</i>	97
6.2.	<i>Pruebas realizadas</i>	100
6.3.	<i>Resultados obtenidos</i>	102
6.3.1.	Prueba 1 (Cruces por 128).....	102
6.3.2.	Prueba 2 (Histogramas)	103
6.3.3.	Prueba 3 (Media de colores con colores homogéneos)	103
6.3.4.	Prueba 4 (Media de colores con colores no homogéneos)	104
6.3.5.	Prueba 5 (Colores más frecuentes con colores homogéneos).....	105
6.3.6.	Prueba 6 (Colores más frecuentes con colores no homogéneos).....	106
6.3.7.	Prueba 7 (Detección de marcas en entorno simple)	107
6.3.8.	Prueba 8 (Detección de marcas en entorno complejo)	110
6.3.9.	Prueba 9 (Detección de marcas no homogéneas)	113
7.	Conclusiones.....	117
8.	Trabajos futuros	120
	Glosario de términos	122
	Bibliografía.....	123

ÍNDICE DE TABLAS

Tabla 1: Clasificación de sensores.....	20
Tabla 2: Modelo de tabla de requisitos.....	57
Tabla 3: Requisito REQ-001.....	58
Tabla 4: Requisito REQ-002.....	58
Tabla 5: Requisito REQ-003.....	59
Tabla 6: Requisito REQ-004.....	59
Tabla 7: Requisito REQ-005.....	59
Tabla 8: Requisito REQ-006.....	59
Tabla 9: Requisito REQ-007.....	60
Tabla 10: Requisito REQ-008.....	60
Tabla 11: Requisito REQ-009.....	60
Tabla 12: Requisito REQ-010.....	61
Tabla 13: Requisito REQ-011.....	61
Tabla 14: Requisito REQ-012.....	61
Tabla 15: Requisito REQ-013.....	62
Tabla 16: Requisito REQ-014.....	62
Tabla 17: Requisito REQ-015.....	62
Tabla 18: Requisito REQ-016.....	63
Tabla 19: Requisito REQ-017.....	63
Tabla 20: Comparativa entre las técnicas utilizadas.....	118

ÍNDICE DE FIGURAS

Figura 1: Planificación real del proyecto	12
Figura 2: Ejemplo de robot con locomoción por ruedas (Pioneer 3 DX)	18
Figura 3: Arquitectura de control de un robot móvil.	22
Figura 4: Esquema de un navegador autónomo	23
Figura 5: Esquema de un sistema de visión artificial.....	26
Figura 6: Fases de un sistema de visión artificial	28
Figura 7: Esquema de color HSL.....	38
Figura 8: Esquema de un sistema atencional.....	42
Figura 9: Ejemplo de una aplicación de robot.....	47
Figura 11: Robot PIONEER 3-DX.....	52
Figura 12: Ciclo de vida en cascada.....	56
Figura 13: Arquitectura de la aplicación	64
Figura 14: Diagrama de secuencia de la aplicación	67
Figura 15: Fase de detección.....	69
Figura 16: Entorno de ejemplo.....	71
Figura 17: Salida esperada del entorno de ejemplo	71
Figura 18: Cruces por cero sobre una señal	73
Figura 19: Reducción de ruido	78
Figura 20: Ventana deslizante utilizada	79
Figura 21: Ejemplo de marcas no conectadas entre sí.....	80
Figura 22: Marcas no conectadas en la aplicación	81
Figura 23: Esquema de localización utilizado.....	82
Figura 24: Ejemplo de transformación 2-D a 3-D (1)	97
Figura 25: Ejemplo de transformación 2-D a 3-D (2)	98
Figura 26: Visión exterior del entorno	100
Figura 27: Visión desde el robot del entorno	100
Figura 28: Resultados prueba 1	102
Figura 29: Resultados prueba 2	103
Figura 30: Resultados prueba 3	104
Figura 31: Resultados prueba 4	105
Figura 32: Resultados prueba 5	106
Figura 33: Resultados prueba 6	106

Proyecto Fin de Carrera.

Sistema de atención visual para la detección de puntos topológicos de referencia.

Figura 34: Entorno prueba 7	107
Figura 35: Marca 1 de la prueba 7.....	108
Figura 36: Marca 2 de la prueba 7.....	108
Figura 37: Resultados prueba 7	109
Figura 38: Entorno prueba 8.....	110
Figura 39: Marca 1 de la prueba 8.....	111
Figura 40: Marca 2 de la prueba 8.....	111
Figura 41: Marca 3 de la prueba 8.....	111
Figura 42: Resultados prueba 7	112
Figura 43: Entorno prueba 9	113
Figura 44: Marca 1 de la prueba 9.....	114
Figura 45: Marca 2 de la prueba 9.....	114
Figura 46: Marca 3 de la prueba 9.....	115
Figura 47: Resultados prueba 9	116

1. Introducción

1.1. Escenario del proyecto

El desarrollo de robots móviles autónomos es un campo muy en boga hoy en día dentro de la Inteligencia Artificial y más en concreto de la Robótica. Los robots móviles autónomos son máquinas capaces de desenvolverse en un entorno conocido o desconocido, y llevar a cabo diferentes tareas, como por ejemplo la exploración de entornos peligrosos (búsqueda de heridos en situaciones de guerra, búsqueda y desactivación de minas anti-persona), exploración de entornos desconocidos (reconocimiento de terreno en otros planetas, en el fondo marino), asistencia a personas con discapacidad, aplicaciones domésticas y de oficina, etc.

Sin embargo, existe una serie de condiciones iniciales que el desarrollador de software para un robot móvil autónomo ha de solventar antes de comenzar a elaborar servicios más complejos para el mismo. Una de ellas es la localización del robot en un entorno conocido o desconocido. El tipo de navegación de un robot se puede clasificar como geométrica o topológica; en el primer caso, un prerequisite importante para el robot es encontrar un buen método para encontrar la propia posición en el espacio, ya que existen diferentes fuentes de error (por ejemplo, el deslizamiento de las ruedas) que llevan a situaciones en las que el robot tiene almacenada una posición que no es realmente en la que se encuentra en ese momento. La navegación topológica resuelve algunos de los problemas de la navegación geométrica, ya que mejora la localización y la percepción del entorno, mediante la identificación de objetos y elementos del mismo (Mata et al., 2001).

Este trabajo se centra en una de las primeras fases en el desarrollo de la navegación de un robot móvil autónomo: el marcado del terreno o, más conocido por su nombre en inglés, *landmarking*. Una marca de terreno, o *landmark*, es una característica física localizada que el robot puede percibir y utilizar para estimar su propia posición en relación a un mapa que contiene la posición relativa de la marca. Lo que se pretende conseguir con este proyecto es desarrollar un método que permita la asignación de marcas en un entorno desconocido de una manera automática, es decir, la determinación

Proyecto Fin de Carrera.

Sistema de atención visual para la detección de puntos topológicos de referencia.

de qué objetos o regiones del mapa, se pueden considerar como interesantes desde el punto de vista de su posterior localización de manera inequívoca.

Existen muchos métodos para la selección de marcas en un entorno desconocido, entre ellos la aplicación de los modelos de atención. No existe una definición clara de atención, ni tampoco se considera que el mecanismo de atención resida en estructuras o zonas cerebrales claramente diferenciadas en los seres vivos, por lo que existen múltiples clasificaciones de los tipos de atención que se dan en los seres vivos (cada una atendiendo a criterios diferentes). Por ejemplo, si la clasificación se realiza en función del objeto al que se dirige la atención, tendríamos atención sensorial (provocada por un estímulo externo), o intelectual (focalización mental sin influencia externa); si la clasificación se basa en el interés que causa la atención tenemos atención inmediata (hacia un objeto interesante *per se*) o derivada (cuando un objeto es interesante por su asociación con otros); finalmente también se puede clasificar la atención por el modo en que es controlada, existiendo la atención pasiva o involuntaria, y la activa o voluntaria (Pilar Bachiller, 2008).

Existen muchos y muy diversos trabajos sobre la elaboración de modelos computacionales que permitan la representación abstracta de las características que componen los sistemas de atención en los seres vivos para, de esta manera, poder aplicarlas en sistemas artificiales. En este proyecto se realizará un análisis de los modelos computacionales de atención más utilizados y más fiables, para implantar uno de ellos en un servicio automático de detección de puntos topológicos de referencia como parte de un sistema de control de un robot móvil autónomo.

Este proyecto se centrará en la posibilidad de implementar dicho servicio software en el robot móvil *Pioneer 3-DX*, que será descrito y analizado en el capítulo 3.5 de este documento.

1.2. Objetivos

Tal y como se ha esbozado en el apartado anterior, los objetivos principales de este Proyecto de Fin de Carrera son:

Proyecto Fin de Carrera.

Sistema de atención visual para la detección de puntos topológicos de referencia.

- El estudio de los métodos tradicionales de marcado topológico y su aplicación en robots móviles autónomos.
- El estudio de los diferentes modelos computacionales de atención, con intención de analizar los más fiables.
- El análisis de las posibles combinaciones de modelos de atención y métodos de marcado topológico a la hora de implementar un servicio software específico de marcado del terreno en un robot móvil autónomo.
- El análisis del proceso de desarrollo de un servicio software específico sobre un robot móvil autónomo determinado (en este caso, el robot *Pioneer 3-DX*, mediante la simulación por ordenador).
- El proceso de modelado e implementación de un servicio software, basado en visión, mediante la utilización de una cámara web como sensor, que realice una función de marcado topológico en un robot móvil autónomo.
- La realización de pruebas de simulación del servicio software implementado y el análisis de los resultados obtenidos.

1.3. Gestión del proyecto

A continuación (Figura 1: Planificación real del proyecto) se muestra la planificación final seguida en el desarrollo de este proyecto:

Proyecto Fin de Carrera.

Sistema de atención visual para la detección de puntos topológicos de referencia.

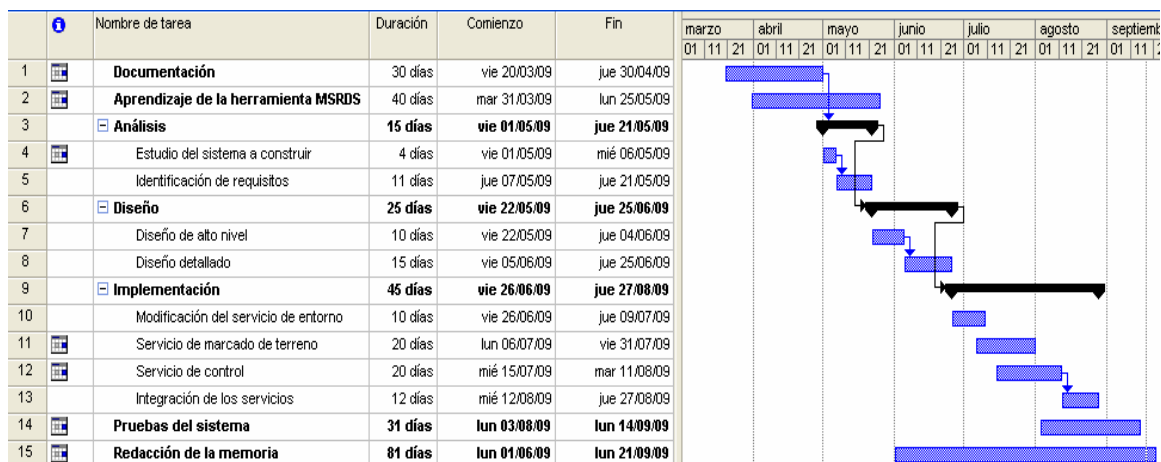


Figura 1: Planificación real del proyecto

Como se puede observar, una de las fases más largas del desarrollo de este proyecto es el aprendizaje de la herramienta que se ha utilizado, Microsoft Robotics Developer Studio (MSRDS), debido principalmente a su complejidad y a la necesidad de adaptarse a conceptos y arquitecturas novedosas.

Aparte de esto, la planificación refleja que se ha seguido aproximadamente un ciclo de vida en cascada, con determinados momentos en los que ha sido necesario revisar o mejorar aspectos de etapas anteriores. Sin embargo, estas eventualidades no han sido frecuentes, por lo que el ritmo de trabajo no se ha visto muy ralentizado.

Proyecto Fin de Carrera.
Sistema de atención visual para la detección de puntos topológicos de referencia.

1.4. Estructura de la memoria

Tras esta introducción, en el presente documento encontraremos un apartado referente al estado del arte en el tema que estamos tratando, es decir, un análisis de los diferentes trabajos existentes, relacionados con la detección de marcas topológicas de referencia (Capítulo 2: Estado del arte).

A continuación se procederá a definir las diversas herramientas software y hardware utilizadas en la realización de este proyecto (Capítulo 3: Herramientas Utilizadas).

Seguidamente, se realizará un análisis de los objetivos fundamentales del proyecto (Capítulo 4: Objetivos), para después entrar en la exposición del trabajo propiamente dicho, en lo que se refiere tanto a las actividades de desarrollo de modelos e implementación software (Capítulo 5: Descripción del trabajo realizado) como a las actividades de prueba del sistema implementado (Capítulo 6: Experimentación y resultados). Finalmente se comentarán las conclusiones obtenidas de la experimentación (Capítulo 7: Conclusiones), y la aplicación del sistema construido en posibles trabajos futuros (Capítulo 8: Trabajos futuros).

2. Estado del arte

En este capítulo se recogerán los estudios más interesantes y significativos realizados hasta el momento acerca de los conceptos fundamentales que se van a utilizar a lo largo del desarrollo del proyecto.

Por tanto, a continuación se expondrán los conceptos más importantes acerca del campo de la robótica cognitiva, los diferentes robots autónomos móviles, sus sensores y los aspectos más importantes de su diseño, los elementos principales de la visión artificial, así como sus etapas más importantes, los estudios realizados sobre la atención visual y su aplicación en sistemas robóticas, y las diversas técnicas analizadas acerca del marcado topológico del terreno o *landmarking*.

2.1. Robótica cognitiva

Según la definición original (Diccionario de la Real Academia Española, 2009), un robot es una máquina o ingenio electrónico programable, capaz de manipular objetos y realizar tareas antes reservadas únicamente a las personas. Sin embargo, aunque la investigación en robótica se ha centrado tradicionalmente en este enfoque, es decir, en la programación de tareas que impliquen un procesamiento de la información llegada a través de los sensores y una planificación directa de acciones consecuentes de dicha información, existe otro enfoque que se centra en intentar dotar a los robots de funciones cognitivas superiores, que les proporcionen la capacidad de razonar, percibir y tomar decisiones de forma autónoma y robusta en entornos desconocidos. Este enfoque es el que se conoce como Robótica Cognitiva.

Aunque las técnicas clásicas de la robótica funcionan correctamente en entornos controlados, en lo que se refiere a entornos complejos y aplicaciones robustas, son necesarios procesos cognitivos superiores. Lógicamente, todos estos procesos que se intentan simular en los robots contruidos artificialmente, tienen una base biológica en la naturaleza, fundamentalmente en los procesos del razonamiento humano, por lo que es las implementaciones de algoritmos basados en la robótica cognitiva pueden suponer avances interesantes en lo que se refiere a la investigación sobre estos campos.

Proyecto Fin de Carrera.
Sistema de atención visual para la detección de puntos topológicos de referencia.

Existen cinco fases fundamentales para un proceso cognitivo, las cuales se enumeran y se explican a continuación (Bernabé, 2007):

- **Sensación:** Consiste en la captación de estímulos por parte de un robot a través de una serie de sensores que pretenden emular diferentes formas de recibir dichos estímulos.
- **Percepción:** Más allá de la sensación, que simplemente recibe los estímulos, la percepción ha de ser capaz de interpretar los valores de dichos estímulos, y transformarlos en información útil para el robot.
- **Atención:** Permite al robot enfocar uno o más “sentidos” en una tarea determinada, o en ciertos elementos del entorno, para centrarse en ellos obviando otra información que no le es útil en ese momento, y recabando únicamente la que puede ser de valor.
- **Memoria:** Consiste en una serie de dispositivos de almacenamiento que guardarán la información recopilada por el robot, para que ésta pueda ser utilizada a largo plazo.
- **Razonamiento inteligente:** El razonamiento se refiere a la elaboración de respuestas lo más adecuadas posibles en función del procesamiento de la información. Cuando se habla de razonamiento inteligente, se está hablando de la capacidad que tiene el robot de adaptarse a diferentes medios, consiguiendo elaborar respuestas más adecuadas.

Uno de los grandes problemas de este campo es la elaboración de una arquitectura de control para la resolución de problemas mediante robótica cognitiva, ya que no existe una solución universal. Una arquitectura de control clásica y más o menos admitida por un gran número de especialistas es la arquitectura de subsunción de Brooks, en la cual las tareas y comportamientos se organizan en capas, totalmente

Proyecto Fin de Carrera.

Sistema de atención visual para la detección de puntos topológicos de referencia.

jerarquizados, de tal forma que las capas inferiores representan comportamientos más primitivos del robot, y tienen prioridad sobre las capas superiores (Brooks, 1991).

Otro enfoque importante a la hora de conseguir robots cognitivos eficaces es la implementación de una buena arquitectura cognitiva. Se denomina arquitectura cognitiva de un robot a la organización de sus capacidades sensoriales y de actuación para generar un repertorio de comportamientos basados en modelos psicológicos. Es decir, es importante conseguir una serie de respuestas bien organizadas y robustas frente a posibles errores, en función de los estímulos que reciba el robot en un momento determinado, para que éste pueda hacer frente a diversos entornos y situaciones.

Una buena arquitectura cognitiva favorece la escalabilidad de la plataforma. Además, en función de la arquitectura seleccionada, existen arquitecturas software que cuadrarán mejor y otras que cuadrarán peor con la misma. Por ejemplo, los sistemas deliberativos clásicos se pueden implementar mejor con programación lógica, mientras que los sistemas basados en comportamientos responderán mejor si se programan de forma concurrente (con varios procesos funcionando en paralelo).

En conclusión, se puede hablar de una alta complejidad a la hora de diseñar robots autónomos, y sobre todo robots que respondan de forma robusta a los estímulos del entorno, fundamentalmente debido a la alta variabilidad que existe en los entornos reales, a la cuál el robot ha de ser capaz de adaptarse de una manera eficaz y eficiente.

2.2. Robots autónomos móviles

La robótica autónoma se refiere a un campo de estudio que se centra en la utilización de robots en entornos a los que el acceso de los seres humanos está muy restringido o es casi imposible (por ejemplo, casos que suponen un riesgo para la salud, o entornos difíciles de acceder). En estos casos, los robots han de ser capaces de ejecutar acciones por su cuenta para conseguir unos objetivos proporcionados a priori. Uno de los objetivos más básicos es el de moverse por el entorno, por lo que será necesario que el robot sea capaz de elegir la mejor ruta, planificarla y navegar por ella.

Proyecto Fin de Carrera.
Sistema de atención visual para la detección de puntos topológicos de referencia.

La planificación del movimiento de un robot autónomo móvil se centra fundamentalmente en los dos siguientes tipos de algoritmos:

- **Globales:** El robot dispone de toda la información del entorno, por lo que es capaz de planificar la forma de moverse y la ruta que va a seguir antes incluso de encontrarse inmerso en dicho entorno. Estos algoritmos requieren de un estudio previo del entorno para conseguir el mapa completo del mismo.
- **Locales:** El robot no dispone de información sobre el entorno o ésta es incompleta. Son los casos en los que el robot va reconociendo el entorno en tiempo real, y no se dispone de ningún mapa del mismo a priori. Existen varios tipos de algoritmos para afrontar este problema:
 - **Heurísticos o reactivos:** El robot únicamente reacciona a estímulos del entorno, sin utilizar ningún modelo del mismo. Por ejemplo, si el robot avanza y llega a una pared cambia de dirección y continúa avanzando hasta llegar a otro obstáculo. Se reduce el cómputo considerablemente, pero tiene más propensión a errores.
 - **Planificación basada en sensores:** Se utiliza también la información que proviene de los sensores, pero intentando planificar el movimiento del robot a priori. Sin embargo, al existir zonas de las cuáles no se tiene información suele ser necesario modificar la planificación inicial a medida que se avanza.

En los siguientes apartados se comentarán cada uno de los puntos clave dentro del campo de la robótica autónoma móvil.

2.2.1. Locomoción

Existen multitud de dispositivos, denominados actuadores, que permiten el movimiento de un robot dentro de un entorno, por lo que a la hora de elegir uno de

Proyecto Fin de Carrera.

Sistema de atención visual para la detección de puntos topológicos de referencia.

ellos, es necesario analizar en profundidad las características del entorno en el que se va a mover el robot, para conseguir una mayor adaptación a dicho entorno.

En general, los mejores resultados los suelen aportar aquéllos dispositivos que imitan a la biología, como por ejemplo juegos de orugas o patas articuladas. Sin embargo, suelen ser altamente complejos de desarrollar, por lo que cuando no existe una necesidad real de utilizar este tipo de dispositivos, las ruedas son las que más se utilizan (como por ejemplo, en el caso del robot que se va a utilizar en este proyecto, el Pioneer 3-DX).



Figura 2: Ejemplo de robot con locomoción por ruedas (Pioneer 3 DX)

A la hora de elegir el sistema de locomoción de un robot autónomo es importante tener en cuenta los siguientes aspectos:

- Estabilidad
 - Puntos de contacto entre el suelo y el robot (numero de piernas, de ruedas, etc.)
 - Centro de gravedad
 - Estabilidad estática o dinámica
 - Inclinação del terreno
- Características del contacto
 - Tamaño y forma de la superficie de contacto
 - Ángulo del contacto

Proyecto Fin de Carrera.
Sistema de atención visual para la detección de puntos topológicos de referencia.

- Fricción
- Tipo del entorno
 - Estructura
 - Medio (agua, aire, suelos blandos, suelos duros, etc.)

2.2.2. Sensores

La utilización de sensores es un aspecto fundamental en los robots autónomos móviles, ya que la forma de recopilar la suficiente información sobre el entorno como para poder realizar una planificación de la forma de actuar del robot, será a través de dichos dispositivos que proporcionarán dicha información. Será necesario analizar toda esa información y entenderla, para poder utilizarla de forma correcta.

Existen diferentes clasificaciones de los sensores, en función del criterio a partir del cuál se quieran clasificar:

- Atendiendo al origen del estímulo:
 - Propioceptivos: Miden valores internos del robot (velocidad del motor, voltaje de la batería, ángulos de los brazos robóticos, etc.).
 - Exteroceptivos: Obtienen información del entorno del robot (distancias, intensidad de luz, etc.), para que el robot extraiga características de dicho entorno.
- Atendiendo al funcionamiento del sensor:
 - Sensores pasivos: Reciben energía ambiental del entorno (intensidad de luz de una cámara).
 - Sensores activos: Emiten energía al entorno para medir la reacción del entorno (láser, sónar, etc.).

La clasificación de los sensores usados en robots móviles (Siegwart, 2004) se expone a continuación. Las siglas significan: A, activos; P, pasivos; P/A pasivo/activo; PC, propioceptivo; EC, exteroceptivo.

Proyecto Fin de Carrera.
Sistema de atención visual para la detección de puntos topológicos de referencia.

Clasificación general	Sensores	PC o EC	A/P
Sensores táctiles	Interruptores por contacto, paragolpes,	EC	P
	Barrido óptico,	EC	A
	Sensores de proximidad.	EC	A
Ruedas / sensores de motor	Codificadores angulares (encoders)	PC	P
	Encoders ópticos	PC	A
	Encoders magnéticos	PC	A
Sensores de dirección (orientación del robot en relación a una referencia inicial)	Compás	EC	P
	Giróscopo	PC	P
	Inclinómetros	EC	A/P
Localización, balizas	GPS	EC	A
	Balizas ópticas o de radiofrecuencia	EC	A
	Balizas por ultrasonidos	EC	A
	Balizas reflectantes	EC	A
Medidas activas	Sensores reflectantes	EC	A
	Sensores ultrasonidos (SONAR)	EC	A
	Medidores láser (LRF)	EC	A
	Triangulación óptica (1D)	EC	A
Sensores de velocidad/movimiento	Radar doppler	EC	A
	Sonido doppler	EC	A
Sensores basados en visión	Cámaras CCD/ CMOS	EC	P
	Cámara PTZ (Pan Tilt Zoom)	EC	A
	Cámara montada sobre brazo robótico	EC	A

Tabla 1: Clasificación de sensores

Como se puede observar en la tabla precedente, existen multitud de sensores que se pueden utilizar, sin embargo, en este proyecto se utilizarán sensores basados en visión, en concreto una Webcam, para la captación de imágenes, y sensores de medidas activas, en concreto un medidor láser, para la determinación de distancias.

2.2.3. Localización de robots móviles

La localización es un problema importante a la hora de programar robots autónomos móviles, ya que se centra en conseguir extraer la posición exacta del robot en un momento determinado, con el menor margen de error posible. Los algoritmos genéricos que se utilizan para la localización estiman la posición en que se encuentra el

Proyecto Fin de Carrera.

Sistema de atención visual para la detección de puntos topológicos de referencia.

robot, a partir de un mapa introducido a priori, y mediante la comparación de dicho mapa con los datos obtenidos de los sensores.

Todas las tecnologías que se puedan utilizar (incluso el GPS) tienen cierto margen de error que, dependiendo de las tareas que ha de realizar un robot, no son aceptables, por los que normalmente es obligatoria la realización de dichos algoritmos de localización. Además, este error se va acumulando a medida que el robot explora su entorno, por lo que aunque en un principio el error cometido puede ser despreciable, cuando el robot lleva mucho tiempo moviéndose por el entorno, el error acumulado puede provocar fallos más graves. Fundamentalmente, la localización implica conseguir una posición relativa del robot respecto de un mapa más que una posición absoluta, ya que de esta manera será más sencillo minimizar el error que se pueda cometer. Dichos errores se producen por múltiples razones, aunque principalmente se deben a medidas erróneas de los sensores (ruido). Por ello, el robot ha de ser capaz de corregir o ignorar el ruido del entorno, lo cual se consigue muchas veces mediante la redundancia (es decir, utilizando varios sensores diferentes para medir lo mismo).

Un algoritmo básico de localización consiste en la realización de análisis de probabilidad, en los que se comparan los datos recogidos por los sensores con el mapa local y se establecen las probabilidades de que el robot se encuentre en un punto del mapa u otro para conseguir, de esta forma, que el robot estima su posición a medida que avanza. Ejemplos de este tipo de enfoque son los filtros de partículas, método muy utilizado en visión artificial, y que consiste fundamentalmente, en evolucionar un conjunto de partículas, que equivalen a posiciones en el entorno analizado, de tal manera que se vayan asociando una serie de pesos o valores a cada partícula, para conseguir que aquellas partículas que más importancia tienen en el conjunto equivalgan a las posiciones del entorno con mayor probabilidad de que el robot se encuentre en ellas.

2.2.4. Planificación y navegación

Mediante la planificación y la navegación, se puede guiar a un robot móvil a través de un entorno hacia un objetivo concreto. Las tareas involucradas en la

Proyecto Fin de Carrera.

Sistema de atención visual para la detección de puntos topológicos de referencia.

navegación se basan en la percepción del entorno a través de los sensores, la planificación de una trayectoria y el guiado del vehículo a través de la misma.

El robot móvil debe poseer una arquitectura de control que coordine los distintos elementos de a bordo (sistema sensitivo, control de movimiento y operación) de forma correcta y eficaz para la realización del objetivo. Un esquema básico de dicha arquitectura de control es el siguiente:



Figura 3: Arquitectura de control de un robot móvil.

Como se puede observar, el control de misión (o de objetivo) coordinará al control de navegación, que determina los desplazamientos, con el control de operación, que interacciona con el entorno de trabajo. El control de misión, por tanto, analizará el problema para resolverlo usando una estrategia, generando como salida un plan de navegación y otro de operación.

El problema de la navegación autónoma se divide en las siguientes cuatro etapas: percepción del mundo (utilizando sensores para crear un mapa), planificación de la ruta (conseguir el objetivo final a partir de subobjetivos), generación del camino (a partir de la secuencia de subobjetivos anterior) y seguimiento del camino.

Un esquema de navegador utilizado en aplicaciones, donde la información acerca del entorno de trabajo varía desde un perfecto conocimiento del mismo hasta poseer un cierto grado de incertidumbre, es el mostrado a continuación:

Proyecto Fin de Carrera.
Sistema de atención visual para la detección de puntos topológicos de referencia.

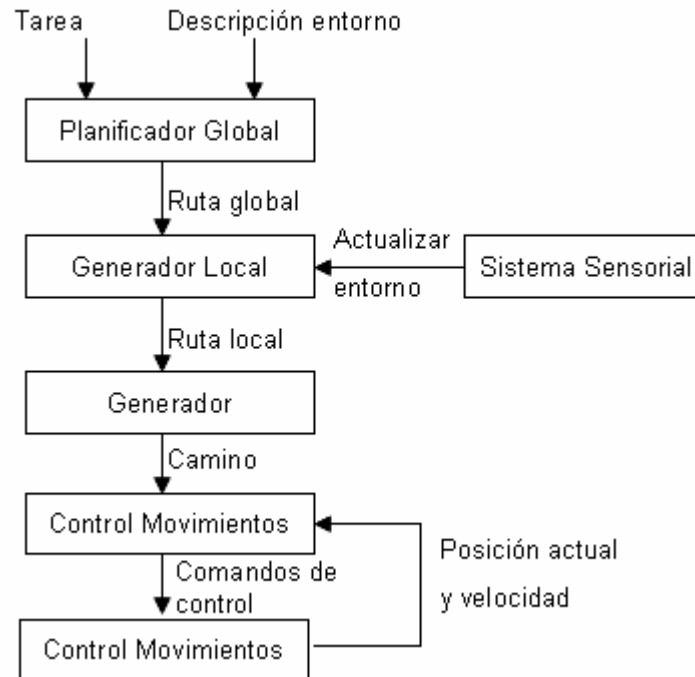


Figura 4: Esquema de un navegador autónomo

La clave del esquema presentado en esta figura es la distinción realizada entre planificación global y local:

- Planificación global: Consiste en planificar la ruta a través de cada uno de los subobjetivos planteados para conseguir una aproximación al camino final seguido, sin tener en cuenta posibles obstrucciones.
- Planificación local: Consiste en determinar la ruta real que se sigue a partir de soluciones propuestas a las diferentes obstrucciones que se encuentran sobre la ruta global.

En el presente proyecto, como se podrá comprobar más adelante, no se hará ningún tipo de navegación global, ya que ésta será completamente aleatoria; sin embargo, sí que se dedicará cierto estudio a la navegación local, para favorecer que el robot se acerque lo suficiente a las marcas de referencia detectadas como para obtener imágenes nítidas de las mismas.

Proyecto Fin de Carrera.
Sistema de atención visual para la detección de puntos topológicos de referencia.

Existen diversas técnicas de planificación, las cuales serán expuestas a continuación:

- Grafos de visibilidad (Nilsson, 1969): Se considera un grafo en dos dimensiones, en el que dos puntos son visibles si y sólo si se pueden unir mediante un segmento recto que no cruce ningún obstáculo en su camino. El grafo por tanto es el resultado de la unión de todos aquellos nodos que son visibles. La ruta que une el inicio del robot con su objetivo final se conseguirá implementando un algoritmo de búsqueda en grafos, por lo que la salida final consistirá en una sucesión de segmentos entre puntos visibles.
- Diagramas de Voronoi: Geométricamente, el diagrama de Voronoi es el lugar geométrico de las configuraciones que se encuentran a igual distancia de los dos obstáculos más próximos del entorno, por lo que esta técnica sitúa la ruta lo más alejada posible de los obstáculos.
- Modelado de espacio libre: Se lleva a cabo mediante el uso de unos elementos geométricos denominados cilindros rectilíneos generalizados (CRG), contruidos a partir de las aristas de los distintos obstáculos del entorno, por lo que con dichos cilindros también se pretende que el vehículo navegue lo más alejado posible de los obstáculos. La ruta final, por tanto, estará compuesta por una serie de CRG interconectados, los cuáles deberán ser recorridos de principio a fin por el robot.
- Descomposición de celdas: Esta técnica se basa en descomponer el espacio libre de obstáculos, en diferentes celdas, para que la ruta final del robot sea una sucesión continua de celdas conectadas. Además de esto, posteriormente será necesario construir un grafo de conectividad que defina la ruta exacta. Una vez que se ha realizado el grafo de conectividad, se implementa un algoritmo de búsqueda en grafos, para conseguir que la ruta siga el grafo desde la celda que contiene la posición inicial hasta la celda que contiene la posición objetivo.

2.3. *Visión artificial*

Se denomina visión artificial a un campo de la Inteligencia Artificial que persigue la extracción de conocimiento y de características a partir del análisis de imágenes adquiridas, generalmente en dos dimensiones. El procesado de la imagen se realiza mediante algún tipo de computador, y se pretende conseguir un análisis de las imágenes similar al que consiguen los seres humanos (que aporte la misma cantidad de información). Existen múltiples sectores en los que es necesario implementar procesos de visión artificial, como por ejemplo, procesos industriales, procesos médicos, automatización de tareas (inspección y supervisión), etc. En lo que se refiere al presente proyecto, la visión artificial va a servir fundamentalmente para la automatización de tareas en lo que se refiere a la detección de marcas topológicas. Es decir, el procesado de las imágenes que el robot reciba a través de la cámara web servirá para la determinación automática de si en la imagen recibida existe o no alguna región que llame la atención del robot como marca de terreno.

Los elementos que componen un sistema de visión artificial son los siguientes (Herrero, 2005):

- Fuente de luz: Proporciona condiciones de iluminación para obtener una imagen clara y nítida.
- Sensor de imagen: es el encargado de recoger las características del objeto bajo estudio.
- Tarjeta de captura o adquisición de imágenes: Interfaz entre el sensor y el computador, que adquiere la información de la imagen.
- Algoritmos de análisis de imagen: Son los algoritmos que se utilizan para procesar la imagen realizando transformaciones sobre la misma, para extraer la máxima cantidad de información posible.
- Computadora o módulo de proceso: Es el sistema que ejecuta los algoritmos definidos anteriormente sobre la imagen obtenida.

Proyecto Fin de Carrera.
Sistema de atención visual para la detección de puntos topológicos de referencia.

- Sistema de respuesta en tiempo real: El sistema de visión artificial puede tomar decisiones en tiempo real en función de los resultados obtenidos, para conseguir una mejora del sistema en el que está implantado.

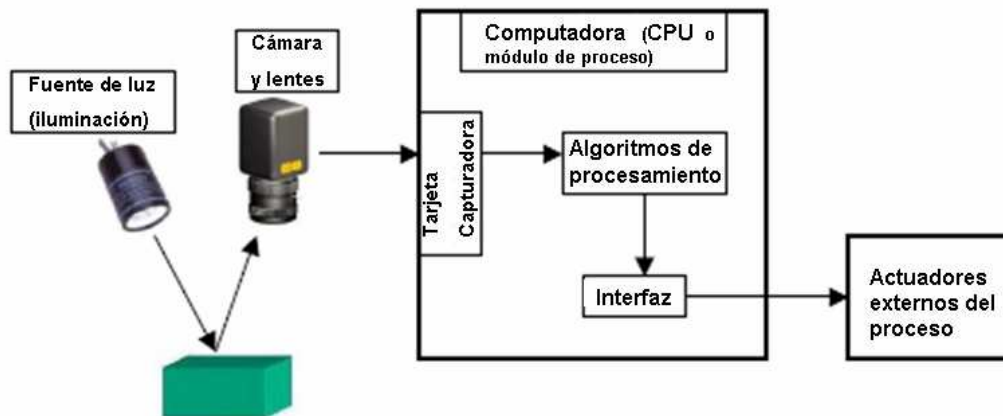


Figura 5: Esquema de un sistema de visión artificial

Existen numerosas etapas en el proceso de visión artificial; a continuación se expondrán las más importantes:

- Filtrado o preprocesado: El preprocesado de una imagen pretende conseguir una versión de la imagen en un formato que sea más fácilmente entendible para su manipulación. En la etapa de filtrado se solucionan problemas derivados del ruido de la imagen, deficiencias en la iluminación, bajo contraste, etc. Dentro de esta etapa, existen diferentes técnicas, como la conversión de niveles de gris (para aumentar la calidad de la imagen), las transformaciones geométricas (para corregir la perspectiva), transformación del histograma (fundamentalmente para la modificación del contraste) y filtrado espacial y en frecuencias (para pequeñas transformaciones de la imagen original).
- Segmentación o aislamiento de objetos de interés: La segmentación persigue principalmente el conseguir separar los objetos que son interesantes de estudiar, dentro de la imagen de que disponemos, de aquellos objetos que suponen ningún interés para el estudio que se está realizando. Ejemplos de esto son la separación de células cancerígenas de otras sanas en imágenes médicas, medición de piezas

Proyecto Fin de Carrera.

Sistema de atención visual para la detección de puntos topológicos de referencia.

(o de partes de ellas) en industria, separación de zonas de terreno en fotografías aéreas, etc. Existen tres técnicas principales para la segmentación, que son: aplicación de umbrales de niveles de gris, agrupación por rasgos comunes y extracción de bordes.

- Extracción y selección de características: En esta fase se extraen las características más importantes de la imagen, teniendo en cuenta que cumplan una serie de criterios, como pueden ser: que sean discriminantes, fiables, que no tengan correlación entre ellas, y que el tiempo de cálculo para extraerlas sea razonable.
- Reconocimiento de formas e inteligencia artificial: La información extraída por un sistema de adquisición de imágenes tras las transformaciones y operaciones realizadas sobre las mismas puede ser representada mediante un vector de características que diferencian a la imagen analizada. Dicho vector puede ser utilizado posteriormente por algoritmos de inteligencia artificial tales como reconocedores o clasificadores, para extraer una serie de conclusiones. Como en todo problema de clasificación será necesaria una fase de selección de características y una de aprendizaje o entrenamiento. En concreto, en el reconocimiento de formas aplicado a la visión artificial se utilizan técnicas como clasificación no supervisada o *clustering*, además de redes neuronales.

A continuación (Figura 6: Fases de un sistema de visión artificial), se ilustran las fases más importantes existentes en un sistema de visión artificial. Hay que tener en cuenta que no todas las etapas tienen por qué ser obligatorias.

Proyecto Fin de Carrera.
Sistema de atención visual para la detección de puntos topológicos de referencia.

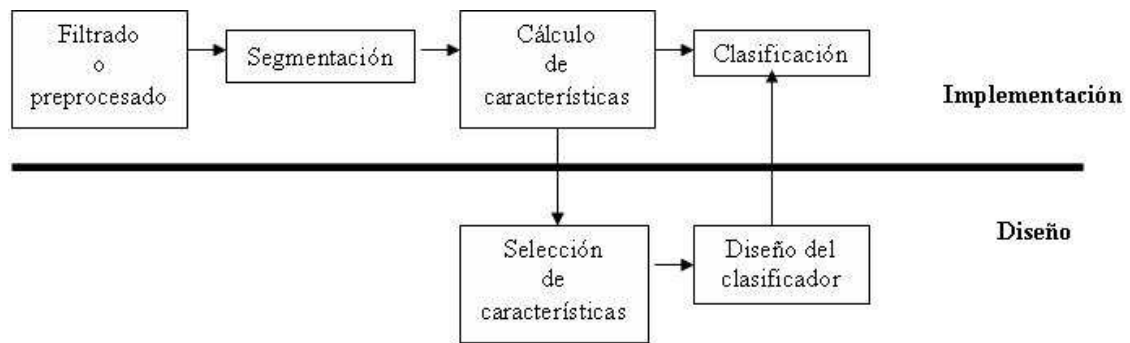


Figura 6: Fases de un sistema de visión artificial

2.4. Atención visual

Existen numerosos estudios en los últimos años sobre la atención visual tanto desde el punto de vista fisiológico como psicológico o neurológico, sin embargo, no se puede encontrar una definición clara y unívoca de la atención, aunque una bastante aproximada es la siguiente: “*La atención es la toma de posesión de la mente, de un modo claro y vívido, de uno entre varios objetos o cadenas de pensamiento simultáneamente posibles.*” (James, 1890).

Como ya se indicaba en la introducción del presente documento, existen numerosos tipos de atención, en función de las clasificaciones que de la misma se han realizado a lo largo de toda la historia:

- Desde el punto de vista del estímulo, la atención se puede dividir en dos tipos: endógena y exógena. En el primer caso, el estímulo procede del propio individuo, por lo que el control depende de las acciones y objetivos del mismo. Este tipo de atención también se denomina *top-down* o guiada por el objetivo. El segundo tipo de atención se produce cuando el estímulo procede del exterior y por tanto el control depende de la capacidad del individuo de captar dicho estímulo, y comprenderlo correctamente para realizar una acción en consecuencia. Este tipo de atención recibe el nombre de *bottom-up* o guiada por el estímulo.
- Desde el punto de vista del sujeto, tendremos igualmente dos tipos de atención visual, en función de si la mirada ha de realizar un movimiento para enfocar el nuevo punto de atención, (atención abierta), o de si dicho punto está contenido en el campo de visión actual y es únicamente un proceso mental el que activa la atención sobre dicho punto (atención encubierta). En muchos casos la atención encubierta es un proceso anterior al de la atención abierta, es decir, nuestra mente capta el nuevo punto de atención sin necesidad de realizar un movimiento de la mirada, pero posteriormente dicho movimiento se realiza para tener un mejor enfoque del punto de atención visual.

Proyecto Fin de Carrera.

Sistema de atención visual para la detección de puntos topológicos de referencia.

- Otras clasificaciones de la atención (Perry y Hodges, 1999) combinan modelos neurológicos para contemplar tres tipos de atención: selectiva y navegante, que se refiere al enfoque de un único estímulo relevante, desestimando otros estímulos, sostenida, en la que se mantiene un único foco de atención durante un largo período de tiempo, y dividida, en la que se comparte la atención entre varios estímulos relevantes.
- Por último, James (James, 1890) propone varias clasificaciones centradas en los aspectos más funcionales de la atención visual:
 - En función del objeto o estímulo: atención sensorial (provocada por la percepción del estímulo), y atención intelectual (proceso mental sin influencia de la entrada sensorial).
 - En función del interés que causa la atención: atención inmediata (provocada por el objeto en sí mismo), y atención derivada (provocada por la asociación del objeto con otros intereses).
 - En función del modo en que se controla: atención pasiva o involuntaria, y atención activa o voluntaria.

En lo que se refiere al trabajo realizado en el presente proyecto, podríamos clasificar la técnica que se va a utilizar en función de los mismos criterios.

Desde el punto de vista del estímulo, se trataría de una atención *bottom-up* o guiada por el estímulo, ya que éste procede del exterior (la marca de terreno), y el control se basa en la capacidad del sistema implementado para detectarlo.

En cuanto al punto de vista del sujeto, se trata fundamentalmente de una atención encubierta, ya que la marca de terreno se encuentra dentro de la imagen que recibe el robot (si es que existe dicha marca), por lo que será un proceso interno el que enfoque la atención sobre dicho punto. Sin embargo, esto se cumple para la navegación global, en la que el robot no busca la marca, sino que navega de manera aleatoria y la encuentra en algunas imágenes, pero no en la navegación local, ya que cuando el robot encuentra la marca, sí que se centra en enfocarla para acercarse a ella, por lo que en ese momento se pasa a una atención abierta.

Proyecto Fin de Carrera.
Sistema de atención visual para la detección de puntos topológicos de referencia.

Si atendemos a los modelos neurológicos, nos encontramos con que se va a aplicar un modelo de atención selectivo y navegante, ya que el proceso se centrará en un único estímulo, desestimando otros que también puedan llamar la atención al robot. En el caso de que existan dos marcas diferentes en la imagen, el robot se centrará únicamente en aquella que se encuentre más cerca.

Por último, atendiendo a la clasificación de James (James, 1890), nos encontramos con una atención sensorial, inmediata, y activa o voluntaria, ya que la atención proviene de la percepción del estímulo externo, y viene dada por un objeto en sí, no por su asociación con otros intereses.

2.4.1. Funciones de la atención

Otro aspecto importante a la hora de hablar de la atención visual, es la función que tiene dicha característica en la vida del ser humano. Una de las teorías más defendidas es la de la función “filtradora” de la atención en lo que se refiere al cerebro humano (Broadbent, 1958; Deutsch y Deutsch, 1963), es decir, debido a la capacidad limitada de la mente humana, es necesario un filtro que evite la sobrecarga de nuestro cerebro con miles de imágenes y pensamientos al mismo tiempo. Esta función desempeñada por la atención, podría ocurrir en las etapas más tempranas de procesamiento mental (Broadbent, 1958), de tal manera que el cerebro analizaría características de todos los estímulos, pero sólo pasaría a estudiar más en profundidad aquéllos seleccionados por el sistema de atención, o podría ocurrir en etapas finales de dicho procesamiento (Deutsch y Deutsch, 1963), en las que, una vez analizados y categorizados en profundidad todos los estímulos recibidos, se selecciona el más importante desde el punto de vista atencional.

Otro punto de vista sobre la función que desempeña la atención en general, y la atención visual en particular en la vida humana, es el que defiende la necesidad de la atención para seleccionar la acción adecuada (Allport, 1987), es decir, la atención como una manera de evitar un desorden en el comportamiento humano de tal manera que la selección de información adecuada nos permita realizar las acciones que más convienen en un momento determinado. Incluso existen teorías que defienden la existencia de rutas

Proyecto Fin de Carrera.

Sistema de atención visual para la detección de puntos topológicos de referencia.

“directas” entre algunos estímulos y determinadas zonas cerebrales (Riddoch et al., 2000), de tal manera que los atributos de dichos estímulos controlan la ruta que hay que seguir para obtener la secuencia de acciones correcta.

2.4.2. Modelos psicológicos y neurológicos de la atención

El gran interés que se ha despertado dentro de los campos de la Inteligencia Artificial y de la robótica acerca de la atención y de sus diferentes manifestaciones, con el objetivo de conseguir robots con comportamientos atencionales lo más similares posibles a los humanos, ha provocado el desarrollo de diversos modelos para entender el funcionamiento de los sistemas atencionales. Aunque los modelos computacionales son más interesantes para el caso que se está tratando, existen también diversos modelos psicológicos, los cuales se expondrán brevemente a continuación, y modelos neurológicos, que se citarán aunque sin entrar en detalle, ya que contienen información que se desvía demasiado del caso que se trata en este proyecto.

Modelos psicológicos de la atención:

- Teoría de integración de características (Treisman y Gelade, 1980): Se observó que el tiempo de respuesta ante un estímulo aumentaba de forma proporcional al número de estímulos de la escena que se está percibiendo, siempre y cuando el objetivo buscado se componga de varias características, mientras que si el objetivo se compone de una sola característica, el tiempo de respuesta no varía por muy alto que sea el número de estímulos de la escena. Esto se explica considerando los diferentes aspectos de la escena visual (color, brillo, orientación, dirección del movimiento) como varios mapas maestros que se obtienen en paralelo antes de prestar atención a algo en concreto. En este caso, la función de la atención sería la de integrar todas estas características divididas en diversos mapas para formar un único objeto, para realizar su búsqueda, definida como ya hemos dicho por varias características.
- Teoría de la búsqueda guiada (Wolfe, 1994): Se trata de una teoría en la cual se definen varios canales que categorizan los estímulos recibidos. Algunos canales

Proyecto Fin de Carrera.

Sistema de atención visual para la detección de puntos topológicos de referencia.

se dedican al cálculo de los niveles de activación que necesitan cada uno de los estímulos, mientras que otros reflejan la activación de un estímulo en función de los objetivos del sujeto. Mediante todos los canales de activación se obtendrá un mapa total de activación, el cuál informará de cuál es el factor o elemento de la escena al que se le va a prestar más atención, y cuál al que se le va a prestar menos (en función de dicho mapa de activación, a mayor activación, mayor atención).

- Modelo “FeatureGate” (Cave, 1999): En este modelo, los mapas atencionales se estructuran jerárquicamente, de tal manera que algunas características se establecen en las capas superiores, y otras se establecen en las capas inferiores. De esta manera, a medida que se asciende en los niveles de la jerarquía, el número de objetivos visuales se reduce, con lo que en el último nivel únicamente queda un candidato, que será el que llame la atención en primer lugar.
- Modelo de redes atencionales (Posner y Delaene, 1994): Se considera la existencia de varias redes atencionales, en lugar de un sistema unitario. Cada red se ocuparía de diferentes aspectos como la detección, la orientación, el control, la consciencia, etc. La jerarquía estaría organizada mediante tres redes: la anterior, que se ocuparía de la detección y selección de objetivos, la de vigilancia, que controla la capacidad del sujeto para reaccionar correctamente en un momento determinado, y la posterior, ocupada en la orientación de la atención (tanto en lo que se refiere al desplazamiento motor como a la atención encubierta).

En este trabajo, aunque no existe un modelo psicológico que se haya seguido de forma completa, se puede apreciar cierta inspiración en el modelo de integración de características, ya que lo que se ha utilizado para el análisis de las imágenes y la detección de zonas que llaman la atención al robot, es el conjunto de características de la imagen, fundamentalmente el color, pero tratándolo con diversas técnicas. Está claro que el modelo de integración de características es mucho más complejo, ya que utiliza multitud de características a la vez en diferentes mapas maestros; sin embargo, existen

Proyecto Fin de Carrera.

Sistema de atención visual para la detección de puntos topológicos de referencia.

similitudes con dicho modelo, en lo que se refiere a la utilización de mapas con regiones que indican mayor o menor relevancia.

Modelos neurológicos de la atención

En cuanto a los modelos neurológicos de la atención, son modelos que contemplan las diferentes estructuras fisiológicas y cerebrales que pueden tener que ver en los procesos atencionales, como son las neuronas y sus conexiones, el córtex visual primario, etc. Como es lógico, son modelos que se escapan al propósito del presente proyecto, ya que implican un mayor conocimiento de conceptos de biología y fisiología.

2.4.3. Modelos computacionales de atención visual

Existen multitud de modelos computacionales de atención visual, la mayoría de ellos motivados por una limitación en los recursos computacionales de las máquinas utilizadas. En general, estos modelos se separan en dos grandes grupos: los que emplean una estrategia de control *bottom-up* (atención guiada por el estímulo) y los que emplean una estrategia de control *top-down* (atención guiada por los objetivos e intenciones del sujeto).

- Modelos Bottom-up:
 - Modelo de Koch y Ullman (1985).
 - Modelo basado en mayor relevancia de Itti y Koch (2000).
 - Modelo STM de Tsotsos (1995).
 - Modelo de Sun y Fisher (2003).
- Modelos Top-down:
 - Modelo de orientación contextual de la atención (2003).
 - Integración de atención top-down en el modelo de relevancia de Itti-Koch (2005).
 - Sistema VOCUS de Frintrop (2005).

Proyecto Fin de Carrera.
Sistema de atención visual para la detección de puntos topológicos de referencia.

Como se puede observar, la oferta de modelos que se pueden aplicar es amplia. Sin embargo, no se puede decir que se haya seguido ningún modelo en concreto para la realización de este proyecto, ya que en general son modelos que superan en complejidad a los objetivos de un proyecto de fin de carrera. No obstante, sí que existen ligeras inspiraciones y similitudes entre algunos aspectos del presente proyecto y alguno de los modelos antes mencionados, teniendo en cuenta que ya se ha señalado anteriormente que el sistema desarrollado ofrece una visión *bottom-up* del problema, ya que se guía por el estímulo.

Por tanto, se puede decir que este proyecto está ligeramente inspirado en el modelo de Koch y Ullman (y por consiguiente con el de Itti y Koch, ya que están estrechamente relacionados), en lo que se refiere a la utilización de características relevantes de la imagen (en este caso, características del color), para conseguir un mapa final de mayor relevancia, en el que se resalten las zonas que llaman la atención.

Sin embargo, como ya se ha indicado, debido a la alta complejidad de aplicar estos modelos computacionales, la relación de los mismos con el presente proyecto se detecta únicamente en ciertos paralelismos, nunca en implementaciones concretas.

2.5. Marcado topológico del terreno (*landmarking*)

El término *landmarking* se podría traducir como “marcado del terreno” y consiste, dentro de la robótica, en la detección de marcas de referencia en el entorno que analiza un robot determinado. Una marca de referencia es una característica determinada del entorno, que por una razón u otra, el robot puede considerar como fácilmente reconocible y unívoca, para su posterior utilización con diferentes fines. Existen diversos ámbitos en los que esta técnica ha sido ampliamente utilizada, como por ejemplo el análisis automático de imágenes médicas, para diferenciar regiones de interés de la imagen de otras que no lo son. Sin embargo, el campo en el que se le ha dado mayor importancia al marcado topológico es el de la localización automática de robots en entornos desconocidos. La técnica se utiliza para que el robot determine su posición relativa con un margen de error lo más pequeño posible, mediante la localización y el reconocimiento de las marcas.

Normalmente, la selección de marcas de terreno se realiza mediante un detector, mientras que la determinación de coincidencias entre marcas se realiza mediante un descriptor. Un detector estable y eficaz es necesario para volver a detectar la misma marca de terreno en vistas diferentes de la misma escena, mientras que un descriptor potente debería capturar fielmente las propiedades de la imagen en la región de interés que se está analizando, de tal manera que se consiga una determinación de coincidencias estable, que ofrezca una alta tasa de detección y una baja tasa de falsa detección.

Existen dos líneas de estudio fundamentales en lo que se refiere al uso de marcas de terreno. La primera de ellas considera como marcas algunas regiones del entorno del robot que pueden ser fácilmente reconocibles posteriormente, aunque no sean necesariamente objetos determinados del entorno. Por el contrario, la otra aproximación utiliza como marcas únicamente objetos del entorno, empleando algoritmos de percepción diseñados específicamente para ellos. Estos objetos pueden ser muros, puertas o pasillos.

La mayoría de estudios realizados sobre el marcado del terreno se centran en marcas topológicas de las cuáles ya se conoce su naturaleza, por lo que la tarea del robot

Proyecto Fin de Carrera.

Sistema de atención visual para la detección de puntos topológicos de referencia.

en estos estudios consiste en explorar el entorno hasta encontrar la marca. Dentro de este tipo de estudios se han desarrollado numerosas técnicas, algunas de las cuáles se expondrán a continuación.

2.5.1. Reconocimiento de patrones

Lo primero que se hace al aplicar un algoritmo de reconocimiento de patrones (Mata et al., 2001) sobre una imagen determinada, es realizar una extracción inicial de las denominadas regiones de interés (ROI en sus siglas en inglés, *Regions of Interest*). Una ROI es una zona de la imagen en la cuál la marca que se está buscando tiene altas probabilidades de ser encontrada.

El proceso de segmentación necesaria para la obtención de las regiones de interés se realiza analizando el espacio HSL de la imagen para posteriormente realizar algunas transformaciones morfológicas sobre la imagen. El análisis sobre el espacio HSL se basa en la utilización de umbrales o *thresholds* sobre los tres componentes de dicho espacio: el componente H (cuya sigla procede de *hue*) se refiere a la percepción del color de un píxel independientemente de las condiciones de iluminación. El componente S (*saturation*), nos indica qué cantidad del color definido por H contiene el píxel analizado, teniendo en cuenta que si $S = 0$, el color del píxel estará dentro de una escala de grises. Por último, el componente L (*luminance*) mide la cantidad de luz absorbida por el píxel, por lo que se puede interpretar como una versión en escala de grises de la imagen.

Por tanto, el algoritmo de segmentación seleccionará aquellos píxeles o regiones de la imagen cuyas componentes HSL superen los límites utilizados como criterio de selección.

Proyecto Fin de Carrera.
Sistema de atención visual para la detección de puntos topológicos de referencia.

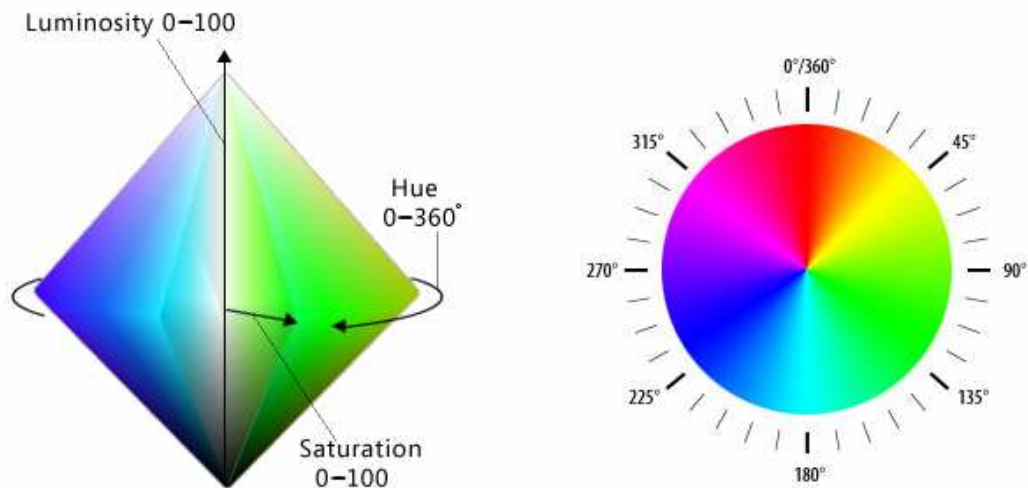


Figura 7: Esquema de color HSL

Una vez conseguida la segmentación de la imagen en regiones de interés, es el momento de realizar la búsqueda de patrones propiamente dicha. Esta búsqueda se puede implementar mediante multitud de algoritmos que acepten o refuten la hipótesis ofrecida por una región de interés de que dicha región es realmente una marca de terreno. Una de las técnicas más usadas en esta fase consiste en la implementación de algoritmos genéticos.

En este caso, el algoritmo que se usa permite reconocer marcas de terreno con perspectivas muy diferentes a la que se está buscando, es decir, elementos de la imagen que, siendo marcas, no lo parecen, debido a que están posicionadas de tal manera que en la imagen que capta el robot aparecen deformadas. Dicho algoritmo utiliza pequeñas ventanas para calcular la correlación entre la marca de terreno propuesta (la hipótesis) y la marca real que se está buscando, en lugar de calcular dicha correlación directamente entre las dos marcas (la real y la propuesta). De esta manera, se reducen mucho más los errores y se pueden encontrar marcas presentes en la imagen con tamaños y deformaciones de perspectiva muy diferentes a la marca real que se busca.

Estos algoritmos genéticos consiguen una rápida convergencia cuando la región de interés estudiada es realmente una marca de terreno (normalmente en una o dos generaciones), mientras que cuando se trata de una falsa hipótesis permite al algoritmo

Proyecto Fin de Carrera.

Sistema de atención visual para la detección de puntos topológicos de referencia.

alejarse rápidamente de esa región de interés y centrarse en otras presentes en la imagen.

Esta técnica demuestra buenos resultados en general; sin embargo, no se han utilizado aspectos de la misma para el presente proyecto, aunque sí existen similitudes: por ejemplo, variando el espacio de color HSV al espacio RGB, usado en este proyecto, sí que hemos utilizado las componentes de color para analizar la información de la imagen. No obstante, no se han utilizado coincidencia de patrones ni algoritmos genéticos en este proyecto. Además, es importante reseñar que esta técnica está más orientada a la detección de marcas topológicas de las cuáles ya conocemos su naturaleza, mientras que el proyecto que nos ocupa está orientado a la determinación automática de qué zonas de la imagen pueden considerarse marcas topológicas de referencia.

2.5.2. Detección de cuadriláteros por agrupamiento de aristas

Esta técnica (Hayet et al., 2002) se utiliza para detectar marcas de terreno con forma cuadrangular, muy presentes sobre todo en entornos artificiales (edificios, oficinas, pasillos, habitaciones, etc.). En este tipo de entornos suele ser muy interesante localizar estas marcas, ya que nos pueden indicar la presencia de ventanas, puertas o cuadros que nos permitan posicionarnos con mayor rapidez.

Es lógico pensar, cuando queremos extraer los cuadriláteros presentes en una imagen, que será muy útil inicialmente extraer todas las aristas posibles, lo cuál se realizará mediante técnicas clásicas de extracción de contornos y de segmentación de la imagen. Una vez hecho esto, habrá que elaborar un algoritmo de coincidencias, primero con las aristas orientadas horizontalmente, y posteriormente con las orientadas verticalmente.

El proceso de extracción de la marca de terreno tiene varias fases:

1. Inicialmente, se realizará una extracción de parejas de segmentos basada en criterios geométricos. El primer criterio se basa en que una pareja de segmentos (i, k) ha de

Proyecto Fin de Carrera.

Sistema de atención visual para la detección de puntos topológicos de referencia.

tener unos ratios de longitud y de superposición determinados para considerarse una pareja candidata a pertenecer al mismo cuadrilátero. El segundo criterio geométrico defiende que para una pareja (i, j) de segmentos orientados verticalmente ha de existir una pareja (k, l) de segmentos orientados horizontalmente, de tal manera que el sector angular delimitado por las líneas rectas que contienen a los segmentos i y j contiene a los segmentos k y l, y viceversa.

2. La segunda fase se basa en criterios de luminosidad, al igual que se ha explicado en la técnica anterior, de tal manera que se extraen perfiles de luminosidad para cada segmento encontrado, y se desechan aquéllos que no cumplen unos criterios determinados (por ejemplo, la superación de determinados límites o *thresholds* en la luminosidad de los píxeles que componen el segmento analizado).
3. Por último, se establecen criterios de unicidad y convexidad entre parejas de segmentos. La regla de unicidad implica que para dos parejas de segmentos, cualquier segmento de la primera pareja puede estar asociado por lo menos a un segmento de la segunda pareja. La regla de convexidad establece que dos parejas de segmentos definen dos cuadriláteros que han de cumplir la regla de inclusión plena o, por el contrario, la regla de no intersección.

Los algoritmos utilizados en esta técnica no han servido para la realización del presente proyecto, ya que están basados en el análisis de formas presentes en la imagen más que en el análisis de color. Sin embargo, sería interesante tener esta técnica en cuenta si se piensa en futuras ampliaciones de este trabajo.

2.5.3. Marcas de terreno atencionales

Existen estudios realizados acerca de la extracción de marcas de terreno utilizando sistemas atencionales (Frintrop y Jensfelt, 2008), los cuales se asemejan al presente proyecto, que se podría definir como la implementación de un sistema atencional sencillo. En el estudio que nos ocupa en este apartado, se utiliza un sistema doble, con una parte *bottom-up* (consultar apartado de Atención Visual) que se ocupa de analizar la relevancia provocada únicamente por el contenido de la imagen que se está

Proyecto Fin de Carrera.

Sistema de atención visual para la detección de puntos topológicos de referencia.

analizando, y una parte *top-down* que considera un conocimiento previo y una información del objetivo a buscar, para que la búsqueda sea más orientada y no tan aleatoria. Sin embargo, la parte interesante de este estudio es la que se refiere al análisis puro de la imagen.

Éste análisis realiza cálculos sobre diversas características de la imagen, en concreto intensidad, orientación y color, de forma independiente, para ser fusionadas posteriormente en un mapa de relevancia atendiendo a la unicidad de las características en la imagen. Esta relevancia se asigna de tal manera que una característica que ocurre de forma aislada tendrá mayor valor que una característica que ocurre frecuentemente.

En el mapa de relevancia, las regiones más llamativas, es decir, las que más captan nuestra atención, serán las señaladas como regiones de interés. Esto se puede conseguir de dos maneras, la primera y más sencilla sería delimitar un rectángulo alrededor del punto máximo o puntos máximos del mapa de relevancia, mientras que una técnica más elaborada consistiría en encontrar la región que corresponde al punto seleccionado, mediante un algoritmo recursivo que encuentra todos sus vecinos con relevancia suficiente como para pertenecer a su región y por tanto a la región de interés de la imagen.

Proyecto Fin de Carrera.
Sistema de atención visual para la detección de puntos topológicos de referencia.

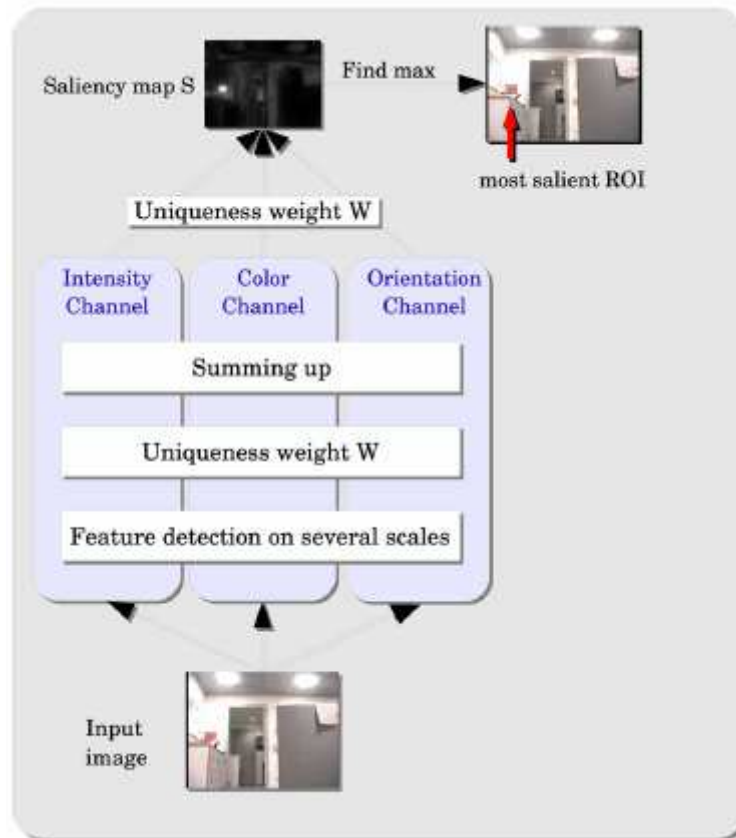


Figura 8: Esquema de un sistema atencional

En la imagen superior (Figura 8: Esquema de un sistema atencional, Frintrap y Jensfelt, 2008) se puede observar cómo la imagen original se divide en tres canales, cada uno de los cuáles analiza una de las características de la imagen (intensidad, color y orientación). Mediante la detección de características en diversas escalas, la asignación de pesos a las mismas, y la fusión de los canales, se obtiene un mapa de relevancia (*saliency map*) en el que el máximo determinará la región de interés y por tanto la marca de referencia que se ha obtenido.

Para seleccionar los detectores de marcas de referencia en imágenes, este método parte de la idea de que normalmente se extraen una gran multitud de características de la imagen, pero sólo unos pocos sirven para los propósitos del marcado de terreno, por lo que es mejor conseguir un sistema que priorice mejor las características a extraer y sólo obtenga las necesarias. Por tanto, bajo esta asunción, se utilizan detectores como el de Harris-Laplace, o el detector SIFT (*Scale-invariant feature transform*).

Proyecto Fin de Carrera.
Sistema de atención visual para la detección de puntos topológicos de referencia.

En el caso de los descriptores, lo más sencillo es utilizar un vector descriptor que contiene los valores de los píxeles pertenecientes a la región, y algunos de las zonas circundantes. Sin embargo, el porcentaje de errores utilizando esta técnica suele ser bastante elevado, por lo que se suelen utilizar otros descriptores más elaborados, como el descriptor SIFT, o el descriptor GLOH (*Gradient Location and Orientation Histogram*).

Es lógico pensar que esta técnica es la que más se asemeja a la utilizada en este proyecto, sobre todo en lo que se refiere a la obtención de un mapa de relevancia sobre la imagen recibida en el que se detallen las regiones que más llaman la atención del robot.

2.5.4. Detección de marcas basada en operadores de color

En esta técnica (Dodds y Hager, 1997), la detección y determinación de marcas se realiza mediante la utilización de características del color. En general existen muchas características que se pueden utilizar dentro de este ámbito, como pueden ser el color dominante, el valor medio del color, u otros aspectos estadísticos fácilmente utilizables, pero en esta técnica se utilizan los histogramas, almacenándose las marcas de terreno como un conjunto de dos histogramas, uno que representa los colores de la marca en sí, y el otro que representa los colores que rodean a la marca.

El algoritmo de búsqueda de marcas de referencia sigue una técnica similar a la explicada anteriormente para determinar la región de interés que representa un píxel seleccionado, es decir, una técnica de crecimiento de región. El algoritmo es el siguiente:

1. Se divide la imagen en un conjunto de regiones de tamaño reducido, para que sea más sencillo trabajar con ellas.
2. Se obtiene el histograma de color de cada una de las regiones en que se ha dividido la imagen, almacenándolo como un vector en el que cada posición nos indica el número de veces que aparece el color que corresponde.

Proyecto Fin de Carrera.

Sistema de atención visual para la detección de puntos topológicos de referencia.

3. Se elige una región como inicial, dependiendo de diferentes criterios, y a partir de ella se analizan sus vecinos. Se considerará una región vecina, como perteneciente a la marca si el producto escalar de los dos vectores que representan los histogramas (uno representando a la región inicial y otro a la vecina) supera un determinado límite establecido como criterio.
4. Si el producto escalar no supera el valor establecido, se considerará que la región no pertenece a la marca, sino a su entorno cercano.

El algoritmo puede finalizar una vez determinada la marca de referencia que contiene a la región establecida como inicial, o puede seguir repitiéndose para encontrar todas las marcas presentes en la imagen. En este segundo caso, el resultado del algoritmo dependerá de cuál es la siguiente región establecida como inicial.

Partes de esta técnica (principalmente, la utilización de histogramas como operadores del color) se ven reflejadas en la realización del presente proyecto; sin embargo, el algoritmo de búsqueda de marcas de terreno no se ha utilizado.

3. Herramientas Utilizadas

En este apartado de la memoria se detallarán todas las herramientas software que se han utilizado a lo largo del desarrollo del proyecto. El sistema operativo utilizado ha sido mayoritariamente Windows Vista, aunque también se ha utilizado Windows XP en algunos momentos determinados. El programa principal de desarrollo del proyecto ha sido Microsoft Robotics Developer Studio (MSRDS), utilizando para dicho desarrollo el lenguaje de programación C#. Además de estas herramientas, se han utilizado diversas aplicaciones del paquete Microsoft Office para la realización de la memoria.

3.1. *Microsoft Robotics Developer Studio*

Microsoft Robotics Developer Studio (MSRDS) es un entorno de desarrollo basado en Windows para estudiantes, desarrolladores profesionales o personas que tienen como afición la programación de robots. MSRDS facilita la creación de aplicaciones robóticas y da la posibilidad de desarrollarlas sobre una gran variedad de plataformas hardware (robots).

Las características principales y las ventajas de MSRDS son las siguientes:

- Incluye una herramienta de programación visual que facilita la creación y depuración de aplicaciones robóticas. También permite generar servicios modulares para hardware y software permitiendo a los usuarios interactuar con los robots a través de interfaces basados en aplicaciones Windows o incluso vía Web.
- La herramienta provee servicios en tiempo de ejecución. Esto es posible gracias a la utilización de la biblioteca CCR, la cual se ejecuta sobre .NET, permitiendo el desarrollo de aplicaciones asíncronas. Con una arquitectura de servicios basada en mensajes, se facilita en gran medida el acceso al estado de los sensores y actuadores del robot tanto a través de un navegador Web como programáticamente. Esto se consigue debido a que el entorno está compuesto de

Proyecto Fin de Carrera.
Sistema de atención visual para la detección de puntos topológicos de referencia.

modelos que permiten construir funciones de alto nivel usando componentes sencillos y facilitando la reutilización del código de los módulos, la calidad y la sencilla sustitución del mismo.

- Se trata de un entorno escalable y extensible ya que el modelo de programación puede ser aplicado para una gran variedad de robots, permitiendo a los usuarios transferir lo aprendido e intercambiarlo entre los diferentes robots con distinto hardware. También permite extender la funcionalidad de la plataforma mediante bibliotecas y servicios adicionales con lo que la ejecución en escenarios puede ser desarrollada usando diferentes lenguajes de programación como C#, VB.net y Python.

Microsoft Robotics Developer Studio permite la simulación de las aplicaciones, lo cual tiene los siguientes beneficios asociados:

- La simulación permite desarrollar aplicaciones y probarlas en un entorno simulado sin necesidad de disponer un robot físico para las pruebas, lo cual facilita la comprobación del comportamiento de diferentes robots en un mismo entorno, sin disponer de ellos físicamente. El hardware puede ser caro y difícil de encontrar, por lo que el entorno simulado puede ser la mejor opción a la hora de programar un robot.
- La simulación resulta muy útil porque permite comenzar el desarrollo de un robot desde un diseño básico e ir complicándolo poco a poco, a medida que se estudian los resultados obtenidos.

Microsoft Robotics Developer Studio utiliza la tecnología .NET., la cuál es una capa que se coloca entre el sistema operativo (SO) y el programador, abstrayendo los detalles internos del SO. Más concretamente, Microsoft define la plataforma .NET como un entorno para la construcción, desarrollo y ejecución de servicios Web y otras aplicaciones que consiste en tres partes fundamentales: entorno de ejecución denominado CLR (*Common Language Runtime*), clases de la plataforma (*Framework Classes*) y lenguajes de programación soportados por la plataforma (ASP.NET, C#, IronPython, etc.).

Proyecto Fin de Carrera.
Sistema de atención visual para la detección de puntos topológicos de referencia.

Una de las ventajas que tiene la plataforma .NET es que cualquier lenguaje de programación puede adaptarse y ejecutarse en ella. Además es portable, es decir, está preparado para poder funcionar correctamente en cualquier tipo de máquina. A pesar de que, por el momento, sólo funciona bien en máquinas con SO Windows, también hay implementaciones para Linux o MacOS. Existe un software, llamado Mono, que sirve para desarrollar y ejecutar programas escritos para .NET, además de aplicaciones en servidores Linux, Solaris, etc.

Microsoft Robotics Developer Studio permite interactuar con un robot a través de una aplicación Windows o una aplicación Web, además es una plataforma extensible y escalable y es la única en su género hasta la fecha.

Su estructura se puede separar en dos bloques principales, por un lado un bloque compuesto por herramientas que nos ofrece el propio programa, como son el simulador y el entorno de programación visual (VPL), y por otro lado, la parte del *Runtime*, donde se pueden encontrar todos los componentes relacionados con la concurrencia y la infraestructura de servicios.

Generalmente las aplicaciones de robots están compuestas por entradas (sensores) y salidas (actuadores y motores). Sin embargo, para que éstos cumplan una determinada función es necesario que se orquesten de la manera adecuada, función realizada por el software de control que se encarga de leer las entradas y en consecuencia activar las salidas correspondientes. Un diagrama de ejemplo de este tipo de funcionamiento se muestra en la siguiente figura.

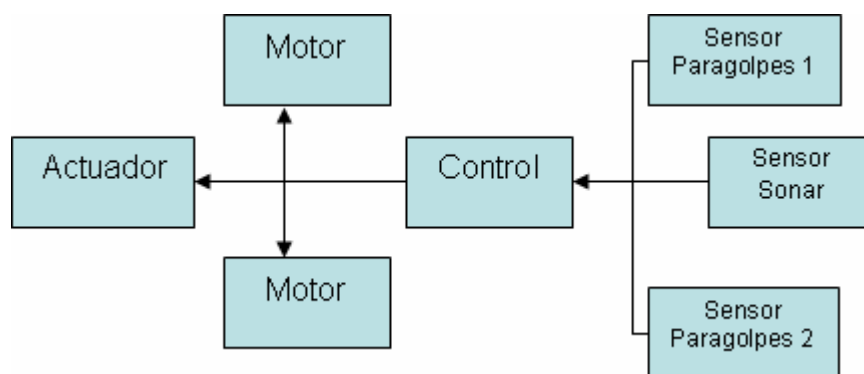


Figura 9: Ejemplo de una aplicación de robot

Proyecto Fin de Carrera.

Sistema de atención visual para la detección de puntos topológicos de referencia.

De acuerdo a lo expuesto anteriormente, vemos que las aplicaciones de robótica que se diseñen deben manejar varios sensores y actuadores de manera concurrente y no secuencial. El problema se complica de forma directamente proporcional al incremento de sensores y actuadores. Microsoft Robotics Developer Studio trata de abordar estos problemas mediante el CCR (*Concurrency and Coordination Runtime*) el cual permite la coordinación de mensajes entre sensores, actuadores y software de control. El CCR está basado en el paso de mensajes de manera asíncrona facilitando la ejecución de los mismos.

Otro elemento importante es el DSS (*Decentralized Software Services*), el cual nos ofrece una interfaz y un protocolo para la definición de servicios descentralizados y la comunicación entre ellos. Por tanto, se puede decir que una aplicación en MSRDS es un conjunto de servicios que se coordinan entre sí.

A continuación se definen los principales conceptos que se aplican en la plataforma MSRDS:

- Soporte de tiempo real (*Runtime*): Se encarga de soportar una gran variedad de hardware de robots conectados directamente a un ordenador personal o robots simulados que pueden ser manipulados en un mundo virtual. Además, el *Runtime* está adaptado para soportar cualquier tipo de aplicación, tales como sensores de entrada, conducción por cable o remota, autonomía del propio robot o cooperaciones entre robots autónomos. El *Runtime* consta de dos componentes principales que facilitan la construcción, supervisión, despliegue y funcionamiento de un amplio rango de aplicaciones. Estos dos componentes se denominan CCR y DSS.
- CCR (*Concurrency and Coordination Runtime*): Permite la coordinación de mensajes y operaciones, abstrayendo al programador del uso de los mecanismos típicos para la gestión de la concurrencia, como son hilos, semáforos, etc. Además plantea un modelo de programa que facilita las operaciones asíncronas explotando el hardware paralelo. Cabe destacar que CCR es un componente

Proyecto Fin de Carrera.
Sistema de atención visual para la detección de puntos topológicos de referencia.

DLL (*Dynamic Linking Library*) autónomo de .NET, accesible desde cualquier lenguaje de programación abarcado por .NET.

- DSS (*Decentralized Software Services*): Combina la arquitectura tradicional Web con conceptos de arquitectura de Servicios Web, para conseguir una arquitectura basada en servicios, muy útil para coordinar aplicaciones distribuidas. Una aplicación es, por tanto, un conjunto de servicios que se coordinan entre sí. El mayor objetivo de la arquitectura es promover la simplicidad y la interoperabilidad, lo cuál se consigue creando composiciones de servicios sin importar si estos servicios están funcionando dentro del mismo nodo o a través de la red. El resultado es una plataforma altamente flexible, capaz de soportar un amplio sistema de aplicaciones. El DSS utiliza los protocolos HTTP (*HyperText Transfer Protocol*) y DSSP (*Decentralized Software Services Protocol*). Hay que decir que DSSP es un protocolo propio que ofrece DSS encargado de la mensajería entre servicios y la creación de servicios descentralizados. Además permite que el estado se mantenga durante el periodo de vida de la aplicación. En cuanto a los nodos DSS, son los encargados de coordinar las actividades de todos los servicios, y pueden estar distribuidos en diferentes ordenadores.
- Servicios: Un servicio es un bloque básico para construir aplicaciones. Los servicios son usados para aislar y crear funcionalidades independientes, donde cada servicio puede comunicarse con otro servicio y con el entorno de ejecución en tiempo real. Los servicios se basan en mensajería DSSP. El protocolo DSSP permite que el estado de un servicio, el cuál contiene el valor de sus parámetros en un momento determinado, se mantenga durante el periodo de vida de la aplicación. Esos valores de los parámetros pueden variar mientras el servicio está arrancado por el mismo servicio o por otros servicios. Dos operaciones importantes que se realizan en todo servicio son las de '*Replace*' y '*Update*'. *Replace* se ejecuta cuando un servicio quiere obtener el estado de otro servicio y *Update* se encarga de obtener solamente los parámetros que han sido modificados en el estado del otro servicio, para de esta manera mantener actualizado el estado del servicio.

Proyecto Fin de Carrera.
Sistema de atención visual para la detección de puntos topológicos de referencia.

- Interacción con otros servicios: Microsoft Robotics Developer Studio permite acceder a la información de un servicio determinado, siendo posible la consulta de los valores de sus parámetros (su estado) o el cambio de algún valor. Esto se puede realizar de una manera muy sencilla desde el propio navegador Web, como se puede comprobar en la siguiente figura:

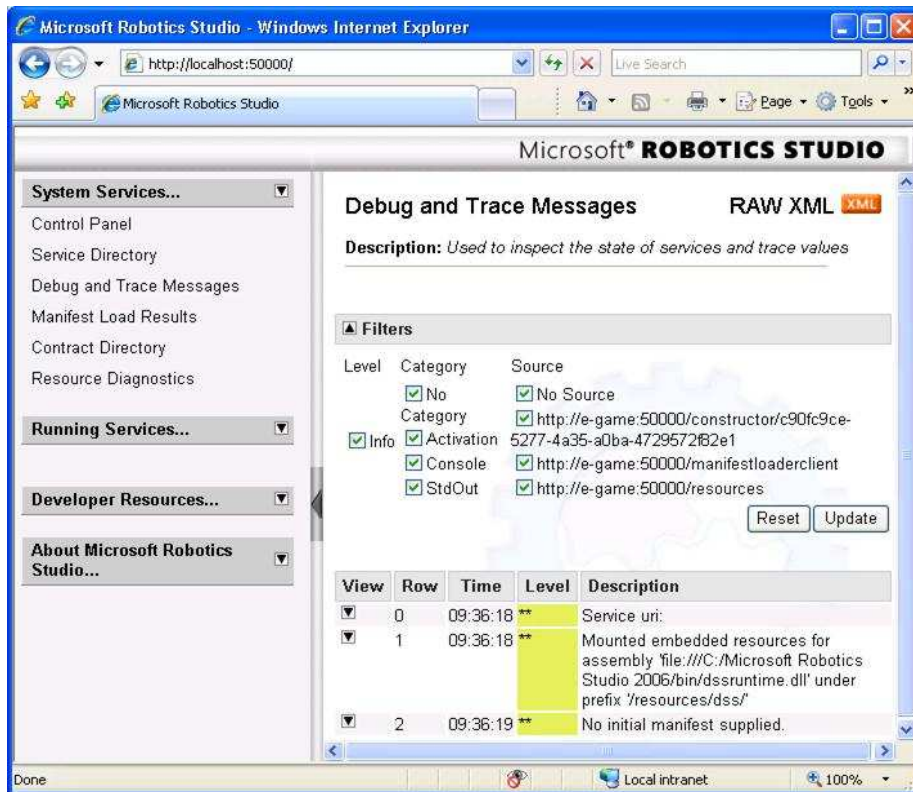


Figura 10: Control de servicios en un navegador WEB

3.2. *Lenguaje de programación (C#)*

La programación de código se ha realizado en el lenguaje de programación C#, para lo cual se ha utilizado la herramienta Microsoft Visual Studio 2008 Professional Edition.

3.3. *Navegador Web*

Microsoft Robotics Developer Studio requiere de la utilización de un navegador web para mostrar tanto los servicios que se están ejecutando en un momento dado como la salida de los mismos, por lo que es necesario que seleccionemos un navegador para realizar las consultas necesarias. En este caso se ha utilizado el navegador Mozilla Firefox 3.0.

3.4. *Microsoft Office*

El paquete Microsoft Office ha sido útil para escribir la presente memoria, ya que existen varios programas de dicho paquete que se han utilizado:

- Microsoft Word 2003
- Microsoft Project 2003
- Microsoft PowerPoint 2003

3.5. *Pioneer 3-DX*

El robot Pioneer 3 DX es una plataforma popular para educación, investigación, creación de prototipos, exhibiciones y otros proyectos. Este robot puede estar equipado con un ordenador integrado en una única placa de formato EBX basada en Pentium, la cual se utiliza para comunicaciones de alto nivel y funciones de control.

Proyecto Fin de Carrera.
Sistema de atención visual para la detección de puntos topológicos de referencia.

Además, el P3-DX también se puede equipar con un ordenador portátil, gracias a que permite una carga de hasta 23 Kg., pero sobre todo gracias a la superficie llana presente en la parte superior del robot. Sin embargo, la desventaja de utilizar esta opción es que se perderá espacio para la instalación de otros sensores y actuadores a bordo.

El robot está basado en un modelo cliente-servidor que proporciona una serie de bibliotecas y utilidades para aplicaciones inteligentes (los robots actúan como los servidores). Normalmente, el P3-DX, al igual que otros productos de MobileRobots, dispone del software específico de alto nivel ARIA (Interfaz Avanzado de Aplicaciones Robóticas), el cuál es un entorno de desarrollo basado en C++ que proporciona soporte para comunicaciones TCP/IP con el robot. Las típicas aplicaciones disponibles a través de ARIA son: mapeo, teleoperación, monitorización, etc. Sin embargo, cuando se utiliza el robot Pioneer 3-DX con la plataforma Microsoft Robotics Developer Studio no se hace uso de este software específico, sino con el software ARCOS, ya que los servicios de MSRDS llaman a la interfaz ARCOS, es decir, el control del robot se realiza, en última instancia, con llamadas a la API nativa.

Las principales características hardware de esta unidad son: comunicaciones basadas en ethernet (opcional), láser (opcional), hasta 252 watios-hora de baterías intercambiables en caliente (las cuales añaden bastante peso al robot), anillo frontal de 8 sensores SONAR, anillo posterior de 8 sensores sonar (opcional), dos motores independientes, 2 ruedas de 19 cm. y una rueda de giro libre. Velocidad máxima de 1.6 m/s. Otras opciones interesantes son los parachoques, pinzas, visión, telémetros, brújula, etc., todos ellos dispositivos que se pueden adquirir del fabricante.



Figura 11: Robot PIONEER 3-DX

4. Objetivos

El objetivo principal del programa que se quiere desarrollar es el de que, dado un entorno desconocido, el robot móvil autónomo PIONEER 3-DX sea capaz de determinar de manera automática qué elementos o regiones de la imagen constituyen una marca topológica de referencia, para una vez detectadas, almacenar la imagen y posición de dichas marcas.

Dicho objetivo principal, por tanto, se dividirá en dos objetivos parciales que se conseguirán de manera secuencial; el primero de ellos consistirá en el análisis de la imagen para la determinación de las marcas topológicas de referencia, mientras que el segundo objetivo se basará en la orientación y acercamiento del robot hacia las marcas encontradas, y la determinación de la posición de dichas marcas. La naturaleza de las marcas que se buscan en el entorno del robot es desconocida a priori, es decir, la aplicación detectará marcas de terreno de forma totalmente automática. Sin embargo, y teniendo en cuenta que el análisis de la imagen para la detección de marcas se basa en análisis de color, es lógico pensar que se detectarán aquellas marcas cuyo color o combinaciones de color llamen la atención en el contexto del entorno que se está analizando.

El servicio principal de la aplicación que se quiere construir será el que se dedique al análisis de la imagen para conseguir un mapa de relevancia que indique, sobre la imagen analizada, cuáles son las regiones de la misma que más llaman la atención del robot; es decir, cuáles son las regiones de la imagen que se pueden considerar marcas topológicas de referencia. Para conseguir esto, se experimentará con diferentes técnicas de análisis de imagen, aplicando conceptos de atención visual, de tal manera que la técnica seleccionada finalmente sea aquella que arroje mejores resultados. Este servicio tomará los datos de la imagen de una cámara Web instalada en el robot, la cuál recogerá las imágenes del entorno para su procesamiento. Sin embargo, el procesamiento real de la imagen se realizará sobre una reducción de la imagen recibida por el robot, ya que es computacionalmente muy costosa la aplicación de técnicas de análisis de imagen sobre una imagen que no haya sufrido un preprocesado.

Proyecto Fin de Carrera.
Sistema de atención visual para la detección de puntos topológicos de referencia.

Finalmente, la estructura de datos que nos indicará qué regiones de la imagen analizada son las que llaman la atención del robot será una imagen binaria en la que cada píxel únicamente podrá tomar los valores 1 ó 0, indicando si la región (o el píxel que la representa) llama la atención del robot o no, respectivamente.

El servicio de marcado enviará sus resultados a otro servicio en el que estará integrado. Este servicio será el de control del robot, y su función principal será variar el movimiento del robot entre una navegación global aleatoria y una navegación local basada en las marcas que detecte el servicio de marcado del terreno. Por tanto, cuando el servicio de control no tenga ninguna marca sobre la que trabajar, el robot se moverá por el entorno de forma aleatoria, únicamente evitando los choques contra las paredes para que el movimiento no se detenga nunca. Sin embargo, en el momento en que el servicio de marcado indique la presencia de una marca topológica de referencia, el servicio de control modificará el comportamiento del robot para conseguir que éste se acerque lo máximo posible a la marca, con el objetivo de almacenar la imagen más nítida posible de la misma. Debido a las comunicaciones necesarias entre el servicio de marcado y el servicio de control del robot, uno de los objetivos de la aplicación es la fácil reutilización de cada uno de los servicios que se implementen.

Para el correcto funcionamiento del servicio de control, será necesario que éste interactúe con el servicio que controla el motor del robot (denominado *DifferentialDrive*) y con un servicio de láser también instalado en el robot. De esta manera, el servicio de control podrá obtener datos de la posición y la orientación del robot, y enviar órdenes al motor del mismo, para conseguir que el robot avance, gire, retroceda o se detenga. Además, el servicio de láser permitirá conocer en todo momento la distancia existente entre el robot y los objetos y obstáculos más cercanos que puede encontrar.

Toda la experimentación del proyecto se realizará de forma simulada, por lo que además de los servicios explicados anteriormente, se hace necesario un servicio que genere el entorno de trabajo y pruebas, así como el robot que se utilizará y su motor, la cámara web y el láser integrado, para que los servicios anteriormente expuestos puedan utilizar todos estos dispositivos. Este servicio que nos sirve para configurar el mundo

Proyecto Fin de Carrera.

Sistema de atención visual para la detección de puntos topológicos de referencia.

virtual en el que se realizarán las pruebas, se nos ha proporcionado, es decir, no forma parte de la implementación del proyecto. Sin embargo, se han realizado ligeras modificaciones sobre el mismo para mejorar el entorno de experimentación. Teniendo en cuenta que el entorno es generado por otro servicio que nos ha sido proporcionado, será necesario que los otros servicios (el de marcado y el de control) se adapten fácilmente a todos los entornos de prueba que se puedan generar.

La salida deseada de la aplicación, por tanto, es un conjunto de imágenes, cada una de las cuales mostrará una marca topológica diferente, y tendrá almacenada de alguna manera (por ejemplo, sobrescrita en alguna parte de la imagen) la posición relativa de la marca, en dos dimensiones.

Lógicamente, todos estos objetivos tienen limitaciones, la mayor de las cuáles consiste en el hecho de que, teniendo en cuenta que el análisis de la imagen se centra en el color, no se detectarán marcas que realmente llamarían la atención por su forma o por otras características. Sin embargo, en el apartado dedicado a trabajos futuros (Capítulo 8: Trabajos futuros), se contemplarán las limitaciones de la aplicación y se propondrán posibles soluciones a adoptar.

5. Descripción del trabajo realizado

En este capítulo se detallará el proceso de desarrollo realizado para conseguir la aplicación especificada anteriormente. Para este desarrollo, la metodología seguida, tal y como se explicó en el apartado de planificación del proyecto (Capítulo 1.3: Gestión del proyecto) ha sido fundamentalmente el seguimiento de un ciclo de vida en cascada. Esto es, la mayor parte del tiempo se ha conseguido un avance más o menos lineal sobre la aplicación (análisis, diseño, implementación y pruebas); sin embargo, ha habido momentos en los que ha sido necesario volver atrás y revisar o mejorar aspectos de etapas anteriores (requisitos, diseño detallado, etc.).

A continuación (Figura 12: Ciclo de vida en cascada) se expone un esquema del ciclo de vida en cascada:

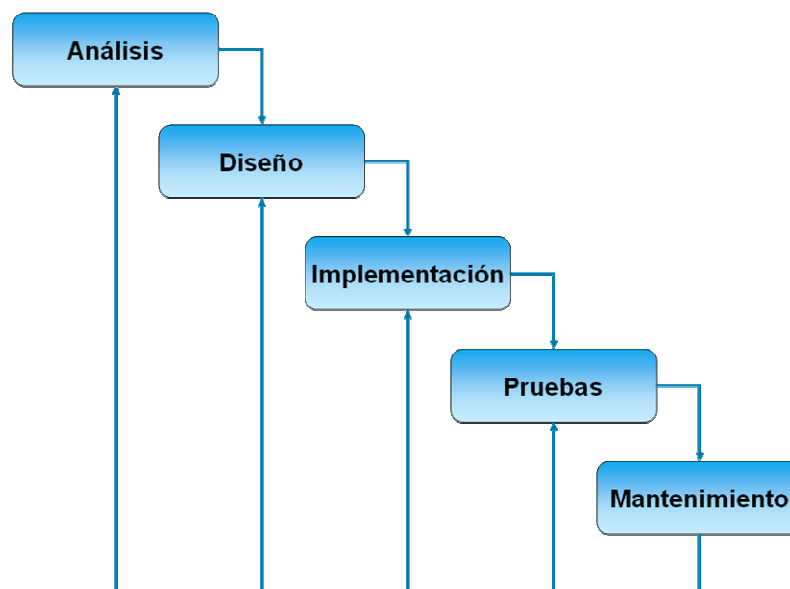


Figura 12: Ciclo de vida en cascada

Lógicamente, en este proyecto se llega hasta la fase de pruebas, ya que no es necesario el mantenimiento de la aplicación. Se puede observar cómo, en cada fase, puede ser necesario volver a una fase anterior para modificar o mejorar aspectos ya desarrollados.

Proyecto Fin de Carrera.
Sistema de atención visual para la detección de puntos topológicos de referencia.

En este capítulo, por tanto, se comenzará exponiendo los requisitos de la aplicación, extraídos del análisis de lo que se quiere conseguir con la misma, y detallados lo máximo posible para evitar ambigüedades. A continuación, se pasará a explicar la arquitectura de la aplicación, con diagramas que describan detalladamente el diseño de alto nivel realizado. Una vez hecho esto, se expondrá el funcionamiento de la aplicación propiamente dicho, centrándonos en el trabajo realizado, las dificultades encontradas, y la explicación de los algoritmos más relevantes del sistema desarrollado. Después será el turno de describir los módulos y clases en los que se descompone la aplicación (el diseño a bajo nivel), explicando aquellas clases y funciones que se consideren más importantes en el funcionamiento de la aplicación. Por último, se incluirá un pequeño manual de usuario en el que se expliquen los pasos necesarios para ejecutar la aplicación y visualizar su funcionamiento.

5.1. Requisitos de la aplicación

A continuación se expondrán los requisitos fundamentales que ha de cumplir el sistema software a desarrollar para conseguir un correcto funcionamiento. En la siguiente tabla (Tabla 2: Modelo de tabla de requisitos) se expone el modelo que se va a utilizar para la descripción de los requisitos de la aplicación:

Identificador: REQ-XX			
Descripción:			
Necesidad:	<input type="checkbox"/> Esencial	<input checked="" type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Prioridad:	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja

Tabla 2: Modelo de tabla de requisitos

Cada campo representa:

- **Identificador:** Nombre unívoco del requisito que se está definiendo. Dicho identificador será del tipo REQ-XX, donde XX será un número consecutivo comenzando en 01.

Proyecto Fin de Carrera.
Sistema de atención visual para la detección de puntos topológicos de referencia.

- **Descripción:** Información básica descriptiva sobre el requisito que se está definiendo.
- **Necesidad:** Nivel de necesidad del requisito. Su valor puede ser “Esencial”, “Deseable” u “Opcional”.
- **Prioridad:** Nivel de prioridad del requisito. Su valor puede ser “Alta”, “Media” o “Baja”.
- **Estabilidad:** Nivel de estabilidad del requisito, es decir, grado de variación que puede sufrir el mismo a lo largo del desarrollo del proyecto. Su valor puede ser “Alta”, “Media” o “Baja”.

Identificador: REQ-001			
Descripción:	Creación de un servicio capaz de detectar imagen y posición de las marcas topológicas de referencia más importantes en el entorno de un robot autónomo móvil.		
Necesidad:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Prioridad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

Tabla 3: Requisito REQ-001

Identificador: REQ-002			
Descripción:	Obtener una imagen más reducida que la obtenida por el robot, para conseguir una mayor velocidad de cálculo, y un mayor realismo al no analizar la imagen por píxeles sino por regiones.		
Necesidad:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Prioridad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

Tabla 4: Requisito REQ-002

Proyecto Fin de Carrera.
Sistema de atención visual para la detección de puntos topológicos de referencia.

Identificador: REQ-003			
Descripción:	Generar una imagen binaria a partir de la imagen obtenida por el robot, de tal manera que se sepa en qué región de la imagen existe atención por parte del robot, y en cuál no.		
Necesidad:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Prioridad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

Tabla 5: Requisito REQ-003

Identificador: REQ-004			
Descripción:	Crear un servicio que sea fácilmente utilizable por otros servicios implementados en el robot.		
Necesidad:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Prioridad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

Tabla 6: Requisito REQ-004

Identificador: REQ-005			
Descripción:	Que el servicio sea fácilmente adaptable a nuevos entornos de trabajo en los que pueda operar el robot.		
Necesidad:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Prioridad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

Tabla 7: Requisito REQ-005

Identificador: REQ-006			
Descripción:	El servicio tiene que ser capaz de funcionar en el entorno de MSRS.		
Necesidad:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Prioridad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

Tabla 8: Requisito REQ-006

Proyecto Fin de Carrera.
Sistema de atención visual para la detección de puntos topológicos de referencia.

Identificador: REQ-007			
Descripción:	Realizar diferentes técnicas de procesamiento de la imagen para obtener las regiones de atención que existen en la misma, de tal manera que se pueda realizar posteriormente un estudio comparativo de dichas técnicas.		
Necesidad:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Prioridad:	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

Tabla 9: Requisito REQ-007

Identificador: REQ-008			
Descripción:	El servicio tiene que poder ser ejecutado en el simulador de MSRS para realizar experimentos en entornos simulados.		
Necesidad:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Prioridad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

Tabla 10: Requisito REQ-008

Identificador: REQ-009			
Descripción:	Los datos de entrada del servicio serán proporcionados por una webcam y por un láser instalados en el robot simulado.		
Necesidad:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Prioridad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

Tabla 11: Requisito REQ-009

Proyecto Fin de Carrera.
Sistema de atención visual para la detección de puntos topológicos de referencia.

Identificador: REQ-010			
Descripción:	Los datos de salida del servicio consistirán en las imágenes de las marcas topológicas de referencia detectadas y sus correspondientes posiciones en dos dimensiones.		
Necesidad:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Prioridad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

Tabla 12: Requisito REQ-010

Identificador: REQ-011			
Descripción:	El servicio tiene que ser capaz de trabajar con diferentes resoluciones de imagen. La resolución de la imagen dependerá del servicio de Webcam, y los valores por defecto serán 320x240.		
Necesidad:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Prioridad:	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

Tabla 13: Requisito REQ-011

Identificador: REQ-012			
Descripción:	El código deberá estar optimizado para minimizar los tiempos de ejecución.		
Necesidad:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Prioridad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

Tabla 14: Requisito REQ-012

Proyecto Fin de Carrera.
Sistema de atención visual para la detección de puntos topológicos de referencia.

Identificador: REQ-013			
Descripción:	El servicio hará uso de bibliotecas externas para la parte de procesamiento de imagen.		
Necesidad:	<input type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input checked="" type="checkbox"/> Opcional
Prioridad:	<input type="checkbox"/> Alta	<input checked="" type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

Tabla 15: Requisito REQ-013

Identificador: REQ-014			
Descripción:	El servicio utilizará la técnica de detección de marcas topológicas que mejores resultados ofrezca de todas las implementadas.		
Necesidad:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Prioridad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

Tabla 16: Requisito REQ-014

Identificador: REQ-015			
Descripción:	El servicio incluirá un sistema de navegación local que permita al robot acercarse a una determinada distancia de la marca detectada para conseguir una mejor imagen y una posición más fiable de la misma.		
Necesidad:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Prioridad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

Tabla 17: Requisito REQ-015

Proyecto Fin de Carrera.
Sistema de atención visual para la detección de puntos topológicos de referencia.

Identificador: REQ-016			
Descripción:	El sistema de navegación global hasta que se detecta una marca en el entorno será aleatorio.		
Necesidad:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Prioridad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

Tabla 18: Requisito REQ-016

Identificador: REQ-017			
Descripción:	El sistema de navegación local una vez detectada una marca irá orientado a conseguir la mejor imagen posible y la posición más fiable posible de la marca detectada.		
Necesidad:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Deseable	<input type="checkbox"/> Opcional
Prioridad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja
Estabilidad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media	<input type="checkbox"/> Baja

Tabla 19: Requisito REQ-017

5.2. Arquitectura de la aplicación

En primer lugar, cabe destacar que se ha intentado seguir siempre un pensamiento modular a la hora de diseñar el sistema, principalmente para facilitar el mantenimiento y la reusabilidad del código, y también porque dicho sistema está desarrollado en el programa Robotics Developer Studio, lo cual nos obliga a desarrollar los servicios de una manera lo más descentralizada y distribuida posible.

Por ello, a la hora de mostrar con un esquema el diseño de nuestro sistema final, éste esquema viene dado por diferentes módulos relacionados entre sí:

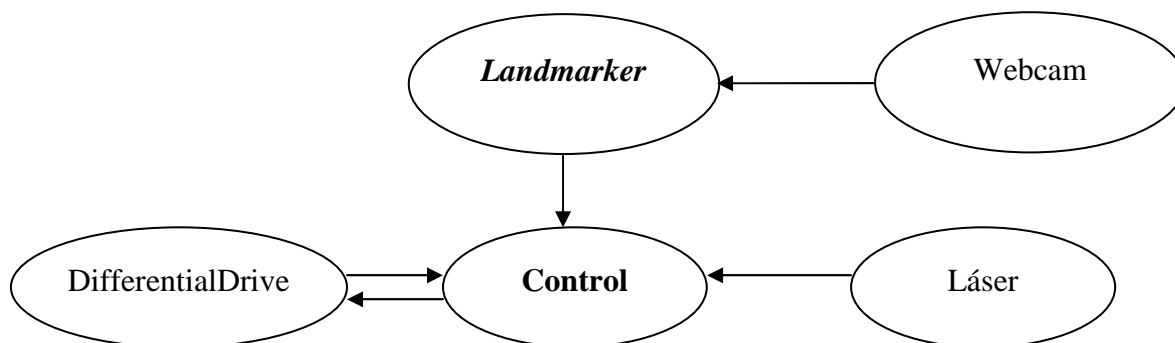


Figura 13: Arquitectura de la aplicación

El esquema muestra claramente cuáles son los módulos (que en este caso representan a servicios) que intervienen en la aplicación. Faltaría por exponer el servicio *Entorno*, que se encarga de generar el mundo virtual por el que se va a mover el robot. Sin embargo, este servicio no ha sido implementado en el desarrollo de este proyecto, y no interviene en lo que se refiere a la funcionalidad concreta de la aplicación, por lo que únicamente es necesario decir que genera los elementos necesarios para el funcionamiento de la aplicación, como son el entorno físico (suelo, cielo, paredes y objetos), el robot y su motor, el láser y la cámara web.

El servicio *Landmarker* es el servicio de marcado del terreno. Su funcionalidad consiste en, a partir de las imágenes que recibe de la Webcam, realizar un análisis de

Proyecto Fin de Carrera.

Sistema de atención visual para la detección de puntos topológicos de referencia.

dicha imagen centrado en el color (con las técnicas que se explicarán más adelante, en el capítulo 5.3: Funcionamiento de la aplicación), y obtener una matriz binaria que representa las diferentes regiones que existen en la imagen, y que indica si una región llama la atención del robot (marcada con un 1 en la matriz binaria), o si por el contrario no contiene elementos interesantes (marcada con un 0 en la matriz binaria). Por tanto, sus datos de entrada son las imágenes que capta la webcam del entorno del robot, y sus datos de salida serán, por una parte la matriz binaria descrita, y por otra parte la imagen original. La imagen original se transmite al servicio de control para que éste vaya disponiendo de todas las imágenes analizadas y sea él el que determine cuándo es suficientemente buena la imagen como para almacenarla como marca de terreno.

La comunicación existente entre el servicio de marcado y la webcam se realizará mediante un iterador dentro del servicio *Landmarker* que, cada determinado tiempo t (definido antes de ejecutar la aplicación en un archivo externo), solicitará una imagen a la webcam.

El servicio de control, por su parte, se relaciona con el servicio de marcado de terreno, con el servicio *DifferentialDrive*, que controla el motor del robot Pioneer 3-DX, y con el servicio de láser LRF (por sus siglas en inglés, *Laser Range Finder*). Este servicio de control será el que se encargue de la navegación del robot mediante el uso del motor y del láser. La navegación global será totalmente aleatoria, de tal manera que cuando el robot avanzará en línea recta, y cuando llegue a una distancia determinada de una pared, girará hacia cualquiera de los dos lados, un número aleatorio de grados, para evitar el choque. En cambio, el servicio de control pasará a navegación local en cuanto reciba del servicio de marcado una imagen que contenga alguna marca. En este momento, la navegación se basará en lograr la mayor aproximación posible a la marca (hasta una distancia mínima definida en el código) para obtener una imagen de la misma de buena calidad. Para ello, girará a derecha e izquierda para centrar la marca en la imagen, y avanzará mientras la marca ocupe poca extensión de la imagen, para acercarse a ella. La comunicación del servicio de control con el de marcado del terreno, por tanto, consistirá en la notificación por parte del servicio *Landmarker* de cada una de las imágenes y su correspondiente matriz binaria (imagen analizada) que genere dicho servicio de marcado. Es decir, la frecuencia con la que el servicio de control reciba cada

Proyecto Fin de Carrera.

Sistema de atención visual para la detección de puntos topológicos de referencia.

notificación del servicio de marcado, será la misma con la que el servicio de marcado solicite cada imagen a la webcam.

Como ya se ha dicho, el servicio de marcado también se comunica con el servicio de láser (de una forma unidireccional) y con el servicio que controla el motor del robot (de forma bidireccional). Las notificaciones que envía el láser al servicio de control son continuas, de tal manera que en todo momento el servicio de control puede saber las distancias existentes a los elementos del entorno más cercanos al robot. Las notificaciones del servicio del motor también son continuas, aunque el robot tiene que estar en movimiento para que se produzcan. Esto es, aunque en un momento el robot deba estar parado, el servicio de control ha de mandar la orden de moverse con una velocidad de 0 m/s al servicio del motor, ya que si no se envía dicha orden, se dejará de recibir información del servicio del motor. Por tanto, el servicio de control envía órdenes al servicio de motor para que avance, retroceda o gire, mientras que el servicio de motor envía información al servicio de control acerca de la posición relativa del robot en un momento determinado, y de la orientación del mismo, teniendo en cuenta que el robot comienza a moverse en la posición (0,0), y con una orientación de 270°, si consideramos que cuando el robot mira a la derecha (según el observador) tiene una orientación de 0°.

Con toda esta información, el servicio de control será capaz de determinar la posición relativa de la marca, en función de la posición relativa del robot, de su orientación, y de la distancia a la marca proporcionada por el servicio de láser, por lo que la salida del servicio de control será un conjunto de imágenes, cada una de las cuáles tendrá sobreimpresa la posición relativa de la marca en dos dimensiones.

A continuación (Figura 14: Diagrama de secuencia de la aplicación) se muestra un diagrama de secuencia con los mensajes y las operaciones más importantes de la comunicación entre el servicio de marcado y el servicio de control:

Proyecto Fin de Carrera.
Sistema de atención visual para la detección de puntos topológicos de referencia.

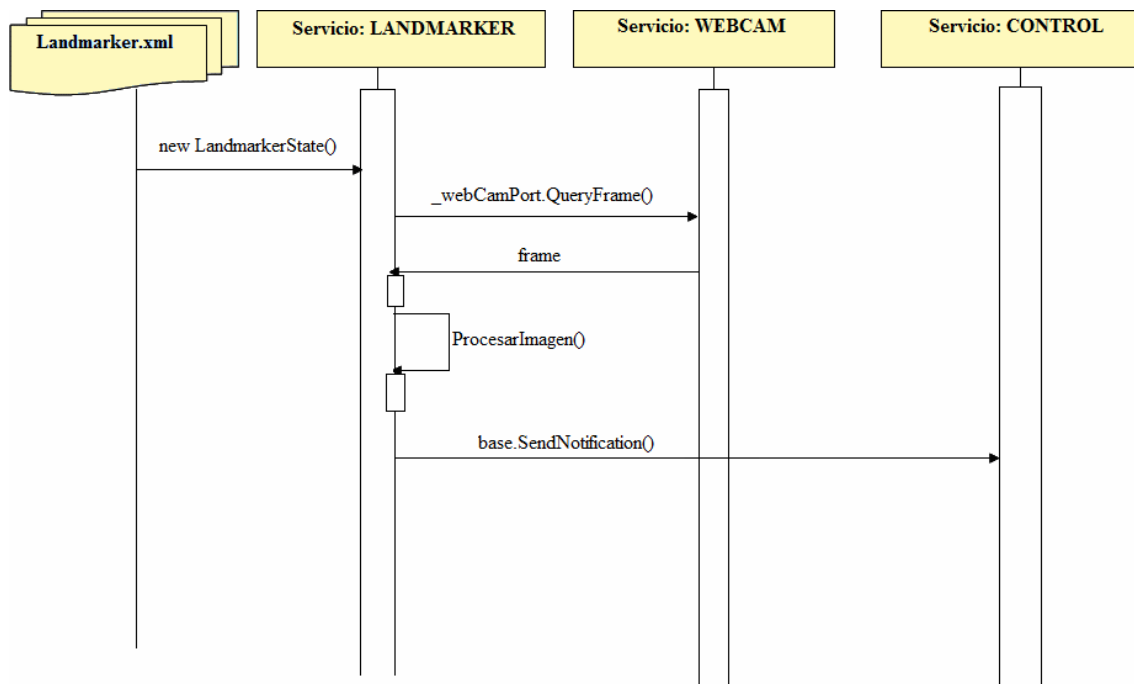


Figura 14: Diagrama de secuencia de la aplicación

Se puede observar que los servicios (de marcado, de webcam, de control, y de láser y *DifferentialDrive*, aunque no aparezcan en la figura) existen desde el inicio de la aplicación, es decir, se entienden como procesos que están funcionando durante todo el tiempo de ejecución de la aplicación. Por tanto, a partir de un fichero externo (que se explicará más adelante, en el capítulo 5.4.3: 5.4.3. Ficheros externos) denominado “Landmarker.xml”, se cargarán los datos relevantes del servicio *Landmarker*, y se generará su estado inicial (en caso de que el fichero XML no existiese, se cargarían los datos por defecto). Este diagrama de secuencia se produce de manera iterativa, cada tiempo t determinado en la configuración del servicio de marcado. Cada vez que se produce, se solicita una imagen al servicio de webcam, mediante el método “QueryFrame” a través del puerto de la webcam (“webcamPort”). La imagen recibida se procesa en el servicio de marcado con el método *ProcesarImagen()*, y se almacena en una estructura de datos junto con la imagen original, para enviársela a todos los servicios suscritos al servicio de marcado (en este caso, el que nos ocupa es el servicio de control) mediante el método “SendNotification”.

Proyecto Fin de Carrera.
Sistema de atención visual para la detección de puntos topológicos de referencia.

Por su parte, el servicio de control recibe datos de los servicios de láser y motor del robot continuamente, y envía también periódicamente datos al servicio de motor para modificar el movimiento del robot.

5.3. *Funcionamiento de la aplicación*

Como ya se explicó en los requisitos expuestos anteriormente, el proceso de obtención de las marcas topológicas de referencia se ha dividido en dos fases fundamentales. La primera fase, que podría ser denominada “fase de detección”, es en la que se centra fundamentalmente este proyecto, ya que es la que contiene todo el proceso de implementación del sistema atencional y de procesamiento de imágenes para la determinación de las marcas topológicas de referencia que existen en las imágenes que recibe el robot. Por tanto, se puede decir que es en esta fase en la que más se ha centrado el desarrollo e implementación del proyecto. La segunda fase, que podríamos llamar “fase de localización”, comprende todos los algoritmos desarrollados para realizar la orientación y el acercamiento del robot hacia las marcas detectadas en la fase anterior, con el objetivo de conseguir la mejor imagen posible, y almacenar la posición de la marca con la máxima fiabilidad.

Al mismo tiempo, la primera fase o fase de detección contiene, tal y como se ha explicado también en el apartado de requisitos, diferentes técnicas para la implementación del sistema atencional y de procesamiento de la imagen, de las que finalmente se ha escogido la que mejores resultados ha proporcionado, para utilizarla en la segunda fase de localización. Dichas técnicas, que se explicarán posteriormente, han sido: “Cruces por 128”, “Histogramas”, “Media de color” y “Colores más frecuentes”.

5.3.1. Fase de detección

En la primera fase, antes de realizar el procesamiento de la imagen es necesario realizar un pretratamiento de la misma, ya que sería altamente ineficiente abordar el procesamiento de la imagen píxel a píxel, debido al elevado número de cálculos que sería necesario hacer. Este pretratamiento también se explicará a continuación, pero antes de ello, se expone un esquema de esta primera fase:

Proyecto Fin de Carrera.
Sistema de atención visual para la detección de puntos topológicos de referencia.

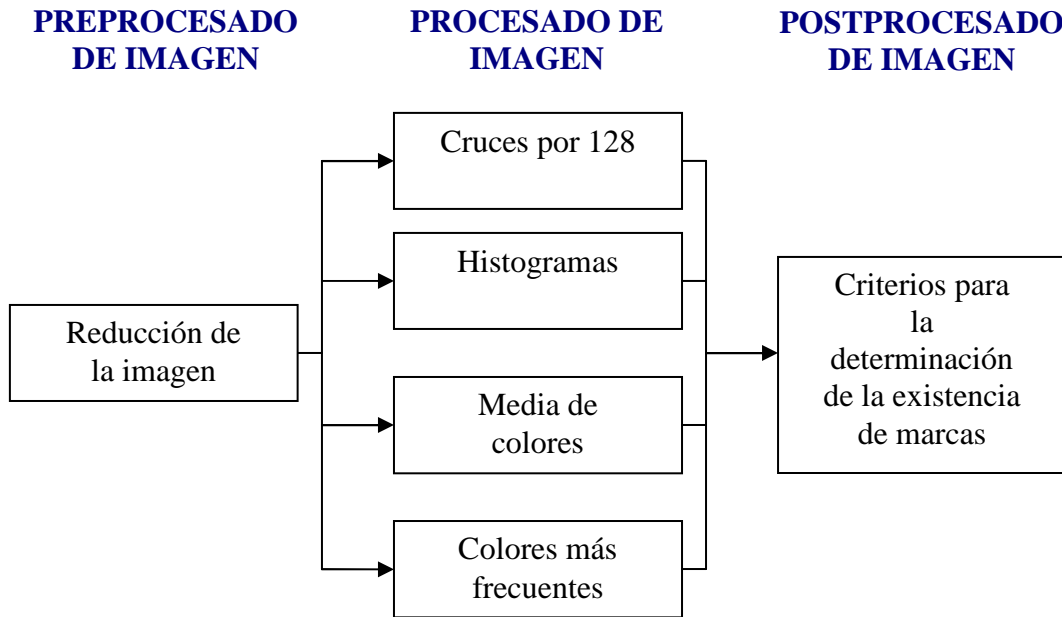


Figura 15: Fase de detección

Se ha optado por efectuar una reducción de la imagen a regiones cuadradas de un número constante de píxeles de ancho y de alto, de tal manera que, en vez de analizar la imagen entera píxel a píxel, se analizará la imagen reducida. Para conseguir dicha reducción, lo que se hace es tomar un divisor del número de píxeles de la anchura de la imagen y un divisor del número de píxeles de su altura. Se divide dicho número de píxeles de ancho por el divisor elegido, para obtener el número de regiones que habrá en el ancho de la imagen reducida. Igualmente, al dividir el número de píxeles de alto de la imagen original por el divisor elegido, se obtiene el número de regiones de alto de la imagen reducida. Es decir, si N_{ancho} es el número de píxeles de ancho de la imagen original, N_{alto} es el número de píxeles de alto, D_{ancho} es el divisor para el ancho y D_{alto} es el divisor para el alto, el tamaño de la nueva imagen, $T_{reducida}$ será:

$$T_{reducida} = (N_{alto} / D_{alto}) * (N_{ancho} / D_{ancho})$$

Además, D_{alto} y D_{ancho} serán el número de píxeles de alto y de ancho de cada una de las regiones, respectivamente.

Por tanto, si por ejemplo tenemos una imagen de 320 * 240 píxeles y elegimos unas regiones de 16 * 12 píxeles, obtendremos una imagen reducida de 20 * 20 píxeles.

Proyecto Fin de Carrera.
Sistema de atención visual para la detección de puntos topológicos de referencia.

Además de esto, dependiendo de la técnica que se vaya a utilizar, en ocasiones será necesario obtener el valor del píxel que representa cada una de las regiones en las que ha quedado reducida la imagen. Este valor se conseguirá simplemente mediante la realización de la media aritmética de cada uno de los componentes RGB de cada uno de los píxeles de la región correspondiente. Es decir, se recorrerán todos los píxeles de la región, y el valor del píxel resultante se obtendrá extrayendo la media de la componente roja de dichos píxeles, la media de la componente verde y la media de la componente azul, para una vez hecho esto, juntar dichos valores en un solo píxel, que representará toda la región.

A continuación se expondrá el desarrollo de las cuatro técnicas de procesamiento de la imagen implementadas para emular sistemas atencionales y así poder determinar qué elementos de la imagen llaman la atención al robot convirtiéndose así en marcas topológicas de referencia. Es importante indicar que todas las técnicas implementadas tienen como objetivo transformar la imagen inicial (o su correspondiente imagen reducida) en una imagen binaria en la que se diferencie únicamente la presencia o no de una marca en cada una de las regiones. Es decir, la imagen final reducida será una imagen binaria en la que un cero indicará que no hay nada que llame la atención al robot en dicha región, y un uno indicará lo contrario, es decir, que existe algo que llama la atención al robot en dicha región, y que por tanto es susceptible de convertirse en marca.

Por ejemplo, si disponemos del siguiente entorno:

Proyecto Fin de Carrera.
Sistema de atención visual para la detección de puntos topológicos de referencia.

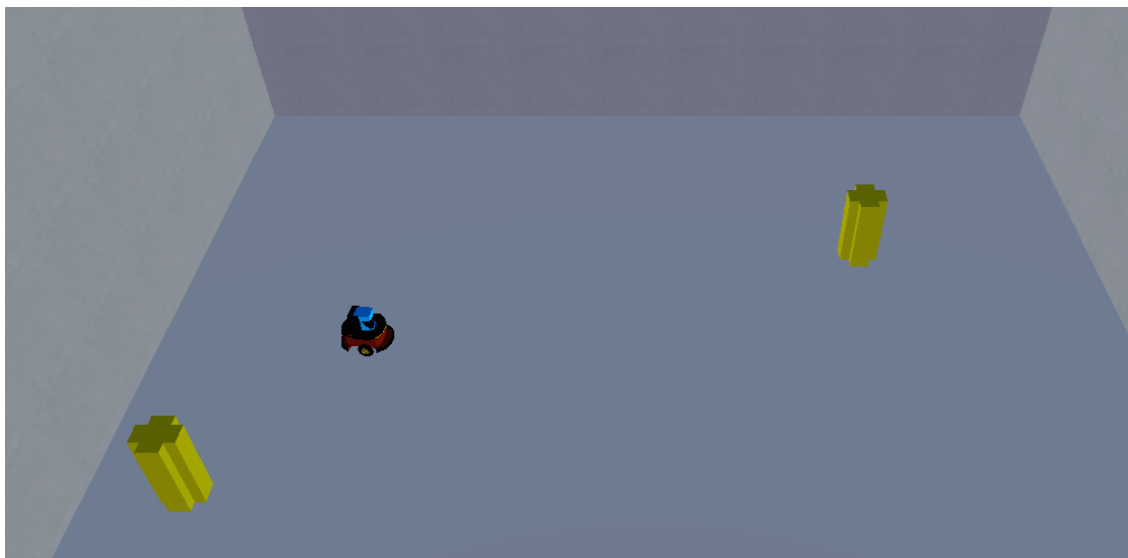


Figura 16: Entorno de ejemplo

Parece lógico pensar que el robot debería determinar como marcas las dos balizas amarillas que se pueden observar en la imagen, por lo que la existencia de una marca debería traducirse en lo siguiente:

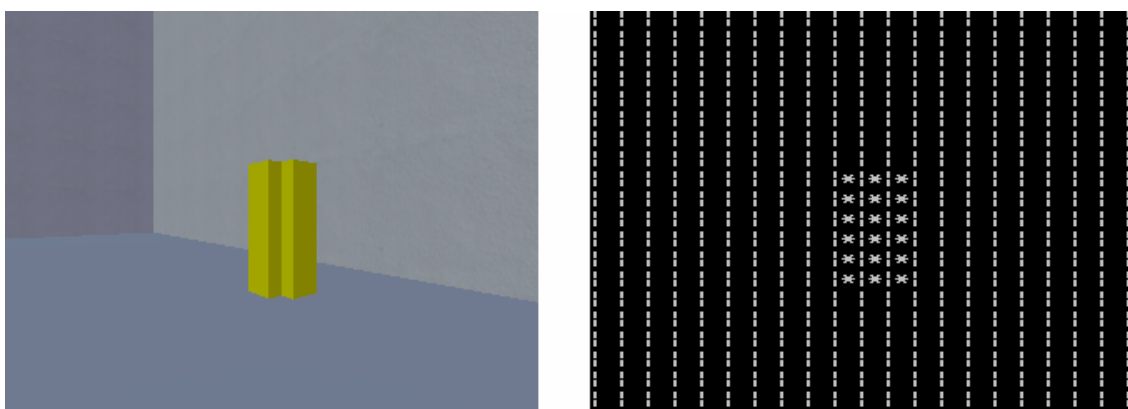


Figura 17: Salida esperada del entorno de ejemplo

Donde la imagen izquierda es la que recibe el robot procedente de la webcam, y la imagen derecha se traduce como una matriz binaria, con sus posiciones separadas por “|”, en la que el símbolo “*” significa existencia de atención, y la ausencia de símbolo significa ausencia de atención. Por tanto, la matriz que podemos observar significaría que todas las regiones marcadas con “*” llaman la atención al robot, por lo que se puede deducir que el conjunto de dichas regiones conexas conforman la marca que sería conveniente almacenar.

Proyecto Fin de Carrera.
Sistema de atención visual para la detección de puntos topológicos de referencia.

Por tanto, se compararán las cuatro técnicas utilizadas, en función de cuáles hayan funcionado mejor para obtener dicha imagen binaria, y bajo el mismo criterio se seleccionará una para pasar a la fase de localización de la marca. Los resultados de dicho estudio comparativo de las técnicas se expondrán en el capítulo 6: Experimentación y resultados.

Cruces por 128:

En muchos sistemas de procesamiento automatizado de todo tipo de señales (auditivas, visuales, etc.), los denominados “cruces por cero” son muy importantes, debido a que suelen proporcionar información, relacionada con la frecuencia, muy valiosa acerca de la señal. Por esta razón se ha decidido implementar una técnica de procesamiento de la imagen basada en dicha manera de procesar la señal. Los cruces por cero consisten en contar, dentro de una señal continua, cuántas veces se cruza el eje de abscisas a lo largo del desarrollo de una ventana de la señal (ya que muchas veces analizar la señal completa sería imposible). De esta manera, se pueden estudiar las variaciones que se producen en el número de cruces por cero a medida que va avanzando la señal, lo cual nos indica el número de veces que la tendencia de la señal ha cambiado.

Por ejemplo, en la siguiente imagen (Figura 18: Cruces por cero sobre una señal) se pueden analizar los cruces por cero existentes en una señal:

Proyecto Fin de Carrera.
Sistema de atención visual para la detección de puntos topológicos de referencia.

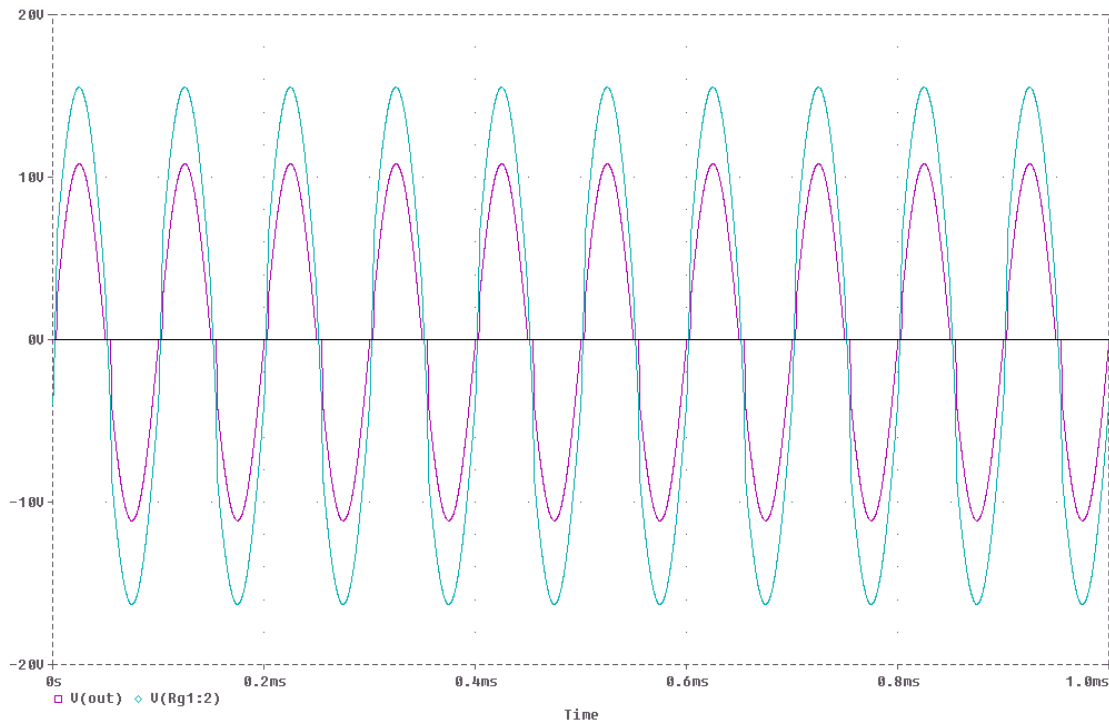


Figura 18: Cruces por cero sobre una señal

Como se puede observar, en la ventana escogida se cuentan un total de 20 cruces por cero, es decir, 20 cambios de tendencia en la señal. Si en una ventana de la misma señal posterior en el tiempo pero de la misma duración, encontrásemos un mayor número de cruces por cero, sabríamos que la frecuencia de la señal ha aumentado y su período ha disminuido, y si encontrásemos un menor número de cruces, sabríamos que la frecuencia ha disminuido, aumentando el período de la señal.

Esta técnica se puede extrapolar al análisis de señales visuales (imágenes), aunque para ello hay que tratar la imagen como si fuese una señal continua, por lo que será necesario realizar ligeras transformaciones.

Una vez reducida la imagen, se ha tratado de considerar cada región de la misma como una señal continua, para lo cual, en cada región de la imagen se han recorrido todos los píxeles y, realizando la media aritmética de los tres valores RGB del píxel, se ha obtenido un valor entre 0 y 255 para dicho píxel, almacenando si dicho valor está por debajo o por encima de 127 (que en realidad sería 128, teniendo en cuenta el 0). Una vez hecho esto, se pasa al píxel siguiente, y si el valor de la media de sus componentes RGB ha cambiado de tendencia (es decir, si antes era menor que 127 y ahora es mayor, o viceversa), se almacena dicho cambio. Al final, lo que se obtendrá será el número de

Proyecto Fin de Carrera.

Sistema de atención visual para la detección de puntos topológicos de referencia.

cambios de tendencia, o cruces por 128, que se producen en cada una de las regiones analizadas de la imagen.

Inicialmente, esta operación se realiza únicamente con el objetivo de ir consiguiendo una media aritmética del número de cruces por 128 que se producen en cada región de la imagen, debido a que se ha considerado necesario disponer de cierta cantidad de información acerca de la imagen, antes de comenzar a valorar si lo que se recibe de la cámara nos indica la presencia de una marca topológica. Para esto, se programó el sistema de navegación del robot de tal manera que, durante el tiempo que tardaba en analizar las primeras n imágenes (donde n es un número especificado en el archivo de configuración, por defecto 20), estuviese dando vueltas sobre sí mismo para intentar captar la mayor cantidad de información posible sobre su entorno. Una vez que se ha almacenado la información de dichas imágenes y se han realizado los cálculos necesarios, ya se puede pasar a comparar las nuevas imágenes con los datos de que se dispone.

Por tanto, a partir de dicha imagen número $n+1$, el proceso de cálculo del número de cruces por 128 de cada región de la imagen será el mismo, sin embargo ahora se realizará una comparación con la media del número de cruces por 128 que se suelen dar en cada una de las regiones de la imagen, la cuál habíamos calculado anteriormente. Se establecerá un *threshold*, es decir, un límite para la diferencia entre el número de cruces por 128 de la imagen actual y el número de cruces por 128 de la media calculada anteriormente, y si el valor de dicha diferencia supera el límite, se considerará que algo ha llamado la atención al robot, y por tanto la región que se está estudiando será marcada.

Histogramas:

Otra de las técnicas más utilizadas en el procesamiento de imágenes para determinar si existe mucha variación entre unas imágenes y otras, y que por tanto puede servir de ayuda en el caso de determinar si existe algún factor o elemento del entorno que llama la atención del robot, consiste en la utilización de histogramas de la imagen.

Proyecto Fin de Carrera.

Sistema de atención visual para la detección de puntos topológicos de referencia.

Los histogramas son gráficos que nos indican, para una imagen determinada, el número de veces que aparece cada uno de los colores que se pueden dar en la imagen, teniendo en cuenta la resolución de color que estamos utilizando. Lógicamente, cuando se analizan imágenes en escala de grises con valores de píxel entre 0 y 255, el histograma de una imagen no ocupará grandes cantidades de memoria, e igualmente se podrán realizar fácilmente cálculos sobre él. Sin embargo, cuando hablamos de imágenes en color como las que estamos tratando en este proyecto, en las que cada componente RGB de la imagen puede tomar valores entre 0 y 255, los problemas de almacenamiento y de procesamiento de los histogramas son mucho mayores, por lo que se hace necesario realizar algunas modificaciones de los histogramas para poder trabajar con ellos más fácilmente.

En este caso, la modificación que se ha realizado ha sido reducir el número de colores a 64, obligando a cada una de las componentes RGB a que sólo pudiesen tomar valores entre 0 y 3 (es decir, dividiendo el valor de la componente entre 64), por lo que se ha conseguido reducir un total de colores de 255^3 a un total de $4^3 = 64$ colores.

Al igual que se ha hecho en el caso de los cruces por 128, se ha dividido el trabajo de calcular el histograma por cada una de las regiones en las que se ha “troceado” la imagen, de tal manera que cuando llega una imagen nueva, para cada una de las regiones en que se divide la misma, se calculará su histograma sobre una paleta de color de 64 valores diferentes.

También como en el caso de los cruces por 128, se ha considerado necesario almacenar una media de los valores de los histogramas a lo largo de las primeras n imágenes que recibe el robot de su entorno, para conseguir un banco de datos suficientemente extenso como para comenzar a trabajar con las nuevas imágenes que vayan llegando a partir de la número n . De esta manera, lo que se hará a partir de dicha imagen será comparar el histograma de la imagen nueva con la media almacenada de las primeras n imágenes. En este caso, la comparación se realizará utilizando dos *thresholds* o límites, uno de ellos para determinar la diferencia mínima entre dos colores iguales de los 64 que componen el histograma utilizado, y el otro para determinar la mínima diferencia entre dos histogramas en general. Es decir, primero se determinarán cuántos

Proyecto Fin de Carrera.

Sistema de atención visual para la detección de puntos topológicos de referencia.

de los 64 colores del histograma de la nueva imagen son diferentes (su diferencia supera el límite establecido) a los equivalentes de la media almacenada, y después se decidirá si el número de colores diferentes supera el segundo límite, lo cuál indicará que existe algo en la región analizada que llama la atención al robot y, por tanto, dicha región será marcada.

Media de colores

La media de colores es probablemente la técnica o el algoritmo más intuitivo que se puede aplicar en el problema que nos ocupa en este proyecto. La idea se basa fundamentalmente en analizar la imagen como perteneciente a un entorno, analizando las diferencias de color directamente con dicho entorno. Por tanto, lo que se hace antes que nada es realizar la reducción de la imagen tal y como se explicó anteriormente para conseguir el valor medio de las componentes RGB de cada una de las regiones en las que se reduce la imagen.

Al igual que en técnicas anteriores, se utilizarán las n primeras imágenes que reciba el robot girando sobre sí mismo para almacenar una media del valor de las componentes RGB en cada región de la imagen reducida para, a partir de la imagen número $n+1$, comparar directamente el valor RGB del píxel que representa a cada una de las regiones de la nueva imagen reducida con el valor RGB del píxel de la misma región en la imagen media almacenada durante las imágenes anteriores.

También de manera equivalente a las técnicas anteriores, se establecerá un umbral, que en este caso consistirá, primero, en saber en cuánto varía el valor RGB de un píxel a otro de los dos comparados, en cada una de sus componentes (roja, verde y azul). Una vez hecho esto, si existen al menos dos componentes de las tres cuya variación supera el límite establecido, se considera que existe un factor que llama la atención al robot, y por tanto se marcará la región.

Proyecto Fin de Carrera.
Sistema de atención visual para la detección de puntos topológicos de referencia.

Colores más frecuentes

La técnica basada en colores más frecuentes es una mejora de la técnica explicada anteriormente, que surge de la necesidad de conseguir una mayor eficacia a la hora de considerar únicamente los colores de la imagen, independientemente del movimiento de la misma. Tal y como se explicará posteriormente en el apartado de “Experimentación”, la eficacia de la técnica “Media de colores” depende en gran medida de que los colores básicos del entorno, es decir, aquéllos que no llamarán la atención del robot, sean lo más homogéneos posible, por lo que se necesitaba una técnica que se centrara únicamente en la diferenciación de colores, sin tener en cuenta la variación, provocada por el movimiento del robot, de dichos colores de una imagen a otra.

Por tanto, lo que se hizo para implementar esta técnica fue, al igual que en la técnica anterior, reducir la imagen recibida almacenando únicamente el valor de las componentes RGB del píxel representativo de cada una de las regiones en las que se dividió la imagen. Esto se hace durante las primeras n imágenes, en las que el robot está únicamente captando información de su entorno. Una vez que se ha realizado este almacenamiento de información, será el momento de determinar cuáles son los colores más frecuentes que se dan en el entorno del robot. Para ello se ha considerado una matriz de 27 colores básicos, de tal manera que se recorrerá la imagen reducida, almacenando a qué color se aproxima más cada uno de los píxeles representativos de las regiones de que se compone la imagen. Una vez hecho esto, será momento de determinar cuáles se consideran como colores más frecuentes, considerando como color más frecuente aquél cuya frecuencia de aparición sea superior a un umbral establecido anteriormente.

Una vez que se han almacenado los colores más frecuentes, dicha información se utilizará a continuación cada vez que el robot reciba una nueva imagen, de tal manera que se analizará cada uno de los píxeles representativos de las regiones de la imagen reducida, y si el color de dicho píxel no está dentro de los colores más frecuentes previamente almacenados, se considerará que dicha región llama la atención del robot, por lo que quedará marcada.

5.3.2. Fase de localización

Una vez que se han conseguido determinar todas las regiones que “llaman la atención” del robot, es el momento de conseguir que dicho robot avance hacia una de ellas para almacenar la imagen más nítida y cercana posible de dicha marca, así como su posición, lo más correctamente posible.

Antes que nada, inicialmente se han eliminado aquellas marcas que podrían significar ruido en la imagen, es decir, marcas que ocupan únicamente una región de la imagen reducida, no existiendo marca en ninguna de las regiones existentes alrededor de la región analizada. Para ello simplemente se barre toda la imagen, analizando si las regiones que hay alrededor de aquélla en la cual existe una marca, están también marcadas. En caso de que ninguna más lo esté, se eliminará la marca, ya que se considerará que existía ruido:

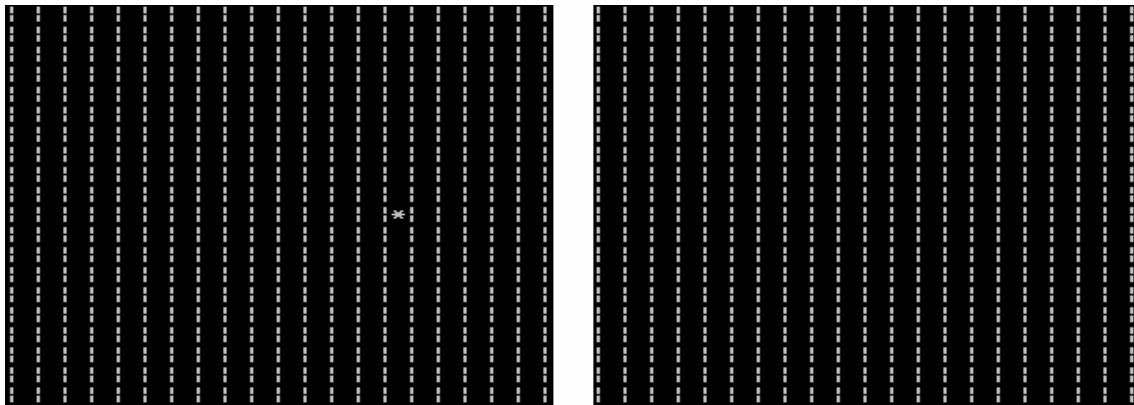


Figura 19: Reducción de ruido

El siguiente paso, una vez que se ha eliminado todo el ruido posible, será analizar la imagen binaria resultante de la detección de marcas para, en caso de que exista más de una marca, decidir hacia cuál se va a dirigir el robot primero, para obtener su imagen. En este caso el criterio que se ha seguido ha sido dirigirse siempre hacia la marca más cercana al robot, es decir, a aquélla marca que ocupe el mayor número de regiones de la imagen reducida. Para conseguir esto, es necesario implementar un algoritmo de etiquetado de componentes conectadas, es decir, un algoritmo que, al mismo tiempo, diferencie cuántos ítems (o componentes conectadas) diferentes existen

Proyecto Fin de Carrera.

Sistema de atención visual para la detección de puntos topológicos de referencia.

en la imagen binaria, y de cuántas regiones se compone cada uno. Dicho algoritmo es el siguiente:

1. Se dispone de un diagrama de vecindad (matriz de tamaño 2x2) que usaremos como ventana deslizante para barrer toda la imagen binaria:

a_{ij}	a_{ij+1}
a_{i+1j}	a_{i+1j+1}

Figura 20: Ventana deslizante utilizada

2. Se comienza a deslizar la ventana anterior en la esquina superior izquierda de la imagen.
3. Para cada posición de la ventana se comprueba si las 4 regiones analizadas por la ventana están marcadas o no.
4. Se dispone de un subconjunto T que almacena todos los ítems localizados hasta el momento (subconjuntos de regiones a su vez). Por tanto, de las regiones marcadas localizadas por la ventana deslizante en su posición actual, se determina si alguna de dichas regiones ya está almacenada en T como perteneciente a un ítem ya localizado. Si existen varias que cumplen dicho criterio, se toma el ítem (subconjunto de T) con el mayor número de regiones de todos los posibles.
5. Todas las regiones marcadas que ha localizado la ventana se incluyen en dicho subconjunto elegido en el paso 3, eliminándose de otros subconjuntos a los que podrían pertenecer anteriormente.
6. Si ninguna de las regiones localizadas por la ventana pertenecían anteriormente a un subconjunto de T , se genera un nuevo subconjunto con todas esas regiones.

Proyecto Fin de Carrera.
Sistema de atención visual para la detección de puntos topológicos de referencia.

7. Si no se ha llegado al final de la imagen (esquina inferior derecha), se avanza un paso la ventana deslizante y se vuelve al paso 3.

Con este algoritmo, por tanto, se conseguirá un conjunto de subconjuntos, cada uno de los cuales almacenará todas las coordenadas de un ítem de la imagen. Por ejemplo, si tenemos la imagen siguiente:

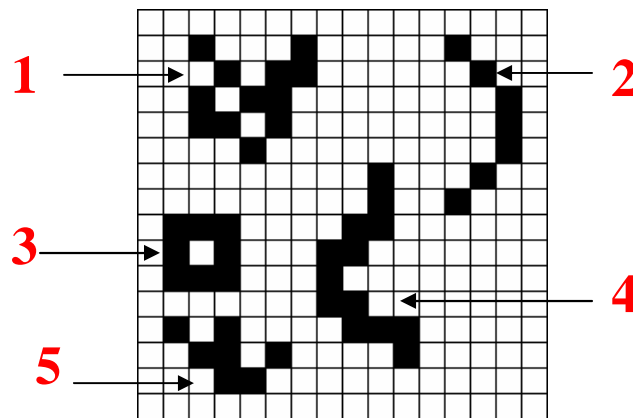


Figura 21: Ejemplo de marcas no conectadas entre sí

Los subconjuntos almacenados serán:

1. (1,2), (1,6), (2,3), (2,5), (2,6), (3,2), (3,4), (3,5), (4,2), (4,3), (4,5), (5,4).
2. (1,12), (2,13), (3,14), (4,14), (5,14), (6,13), (7, 12).
3. (8,1), (8,2), (8,3), (9,3), (10,3), (10,2), (10,1), (9,1).
4. (6,9), (7,9), (8,9), (8,8), (9,8), (9,7), (10,7), (11,7), (11,8), (12,8), (12,9), (12,10), (13,10).
5. (12,1), (12,3), (13,2), (13,3), (13,5), (14,3), (14,4).

Gracias a este algoritmo, conseguiremos determinar cuál es la marca más cercana al robot (o por lo menos, la que más regiones de la imagen reducida ocupa), con lo que podremos acercarnos a ella:

Proyecto Fin de Carrera.
Sistema de atención visual para la detección de puntos topológicos de referencia.

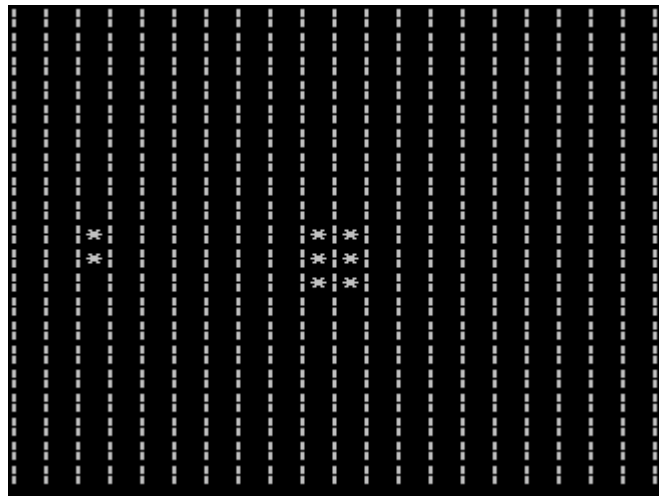


Figura 22: Marcas no conectadas en la aplicación

Como podemos observar, en la imagen binaria resultante de la detección de marcas, obtenemos que existen dos balizas diferentes que le llaman la atención al robot. Sin embargo, la que está a la izquierda la obvia, ya que la que se encuentra a la derecha es más grande o está más cerca del robot, lo cual provoca que esté compuesta por un mayor número de regiones en la imagen reducida. Por tanto, el robot se dirigirá a dicha marca en vez de dirigirse a la que está a la izquierda.

Una vez que se ha determinado a qué marca ha de dirigirse el robot, es momento de manipular el movimiento del mismo para conseguir que la marca esté lo más centrada posible en el momento de almacenar la imagen de la misma. Esto se conseguirá analizando dónde se encuentra la marca dentro de la imagen binaria.

Para ello, teniendo en cuenta que disponemos de todas las coordenadas de la marca que estamos analizando, únicamente será necesario obtener la región situada más a la izquierda y la situada más a la derecha de la imagen. Si se observa que la marca está demasiado a la izquierda de la imagen, habrá que girar el robot hacia la izquierda, y si se observa que la marca está a la derecha de la imagen, habrá que girar el robot hacia la derecha.

Proyecto Fin de Carrera.

Sistema de atención visual para la detección de puntos topológicos de referencia.

Igualmente, existirá un momento en el que será necesario detenerse ya que el robot se encontrará lo suficientemente cerca de la marca como para poder tomar una imagen correcta de la misma, y al mismo tiempo, almacenar la posición de la misma. Para saber cuándo debemos detener el robot para almacenar la imagen, utilizamos el sensor láser instalado en dicho robot, que nos proporciona en todo momento un vector de distancias del cuál es sencillo obtener cuál es la distancia más corta a la que se encuentra un obstáculo del robot. Con dicha información sabremos a qué distancia nos encontramos de la marca que queremos almacenar.

Una vez detenido el robot, la imagen es sencilla de almacenar, ya que estamos continuamente obteniendo las imágenes que capta la cámara instalada en el robot. Sin embargo, tendremos que hacer algunos cálculos para obtener la posición en la que se encuentra la marca. Gracias al motor instalado en el robot (*Differential Drive*), somos capaces de acceder a la posición relativa del robot desde que comenzó a moverse por el entorno (momento en el que se encontraba en las coordenadas (0,0)). Asimismo, podemos obtener la orientación en grados del robot, teniendo en cuenta que cuando comienza a moverse, está orientado hacia abajo según nuestra perspectiva al mirar la pantalla del ordenador.

Con estos datos y conociendo la distancia a la que se encuentra la marca, no es difícil obtener la posición de dicha marca, de la siguiente manera:

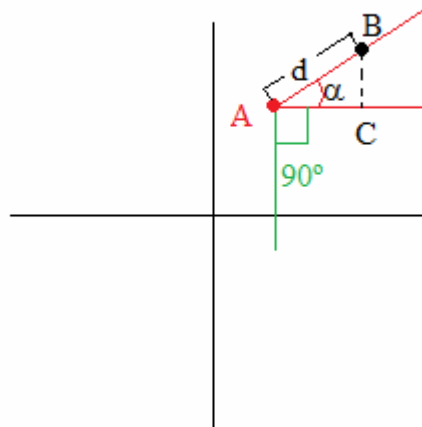


Figura 23: Esquema de localización utilizado

Proyecto Fin de Carrera.
Sistema de atención visual para la detección de puntos topológicos de referencia.

Considerando que el punto A se refiere a la posición del robot, y el punto B se refiere a la posición de la marca, podemos decir que las coordenadas (conocidas) del punto A serán (a_x, a_y) , mientras que las coordenadas (desconocidas) del punto B serán (b_x, b_y) . Sin embargo, sabemos que las coordenadas del punto B se podrán calcular sumando una cantidad a cada coordenada del punto A, es decir:

$$b_x = a_x + m; b_y = a_y + n.$$

Por tanto, nos queda calcular los valores de “m” y de “n”. Otros datos que tenemos son la distancia “d”, que nos proporciona el láser, y el ángulo de orientación del robot, que no es exactamente α , sino que es la suma de α más 90° . Por tanto, cuando obtenemos la orientación del robot únicamente hay que restar 90° al valor obtenido para conseguir el valor de α .

Una vez que sabemos esto, aplicando la trigonometría podemos obtener la longitud del segmento AC, que equivale al valor “m” que hay que sumar a la coordenada horizontal del robot, así como la longitud del segmento BC, que equivale al valor “n” que hay que sumar a la coordenada vertical del robot:

$$AC = d * \text{sen}(\alpha).$$

$$BC = d * \cos(\alpha).$$

Por tanto, las coordenadas del punto B, o lo que es lo mismo, de la marca localizada, son las siguientes:

$$b_x = a_x + (d * \text{sen}(\alpha)).$$

$$b_y = a_y + (d * \cos(\alpha)).$$

Sabiendo que α es la orientación actual del robot menos 90° .

Proyecto Fin de Carrera.

Sistema de atención visual para la detección de puntos topológicos de referencia.

Una vez obtenida la posición de la marca, sólo resta almacenar la imagen y su posición (sobrescrita en la imagen), es decir, la salida del programa. También es importante recalcar que, al tratarse de un sistema totalmente automático, es posible que el robot vuelva a detectar en el futuro la misma marca e intente almacenarla. Sin embargo, para evitar que se duplique la salida, se comprueba si la posición que se quiere almacenar existe ya (con un margen de error) en la lista de marcas que componen la salida del programa. Si dicha posición ya existe, no se almacenará la marca. Además, para evitar que el robot vuelva a detectar la marca inmediatamente después de haberla almacenado, se obliga a dicho robot a realizar un giro de 150°, durante el cuál no analiza las imágenes para ver si existen marcas, por lo que se alejará su atención de la marca antes de volver a analizar las imágenes captadas por la cámara.

Para finalizar, hay que destacar que la fase de detección, así como la eliminación de ruido se encuentran totalmente integradas en el servicio de *Landmarking*, mientras que el resto de la fase de localización se encuentra integrada en el servicio de control del robot.

5.4. Descripción de los módulos y clases

Al tratarse de un proyecto realizado con la herramienta Microsoft Robotics Developer Studio, lo que se ha implementado han sido varios proyectos que generan cada uno un servicio con el nombre del proyecto. Los proyectos que se han implementado se denominan “Entorno”, “Control” y “*Landmarker*”. Además, se han utilizado ficheros externos, los cuáles se expondrán al final del presente apartado.

Los elementos que componen el proyecto *Landmarker* son los siguientes:

- *Landmarker.cs*: Contiene la clase principal, con las operaciones fundamentales para conseguir que el servicio funcione correctamente. Igualmente contiene algunas clases y atributos adicionales, necesarios también para su funcionamiento.

Proyecto Fin de Carrera.

Sistema de atención visual para la detección de puntos topológicos de referencia.

- **LandmarkerTypes.cs:** Contiene las clases, atributos, contratos y miembros utilizados por el servicio durante su ejecución. Contiene el estado del servicio.
- **CameraForm.cs:** Contiene la definición de la pantalla que nos muestra las imágenes que le llegan al robot.
- **Landmarker.manifest.xml:** Manifiesto del servicio. Contiene los datos fundamentales para lanzar el servicio y que comience a funcionar.

Por su parte, el proyecto “Control” está compuesto por los siguientes elementos:

- **Control.cs:** Contiene la clase principal, con las operaciones fundamentales para conseguir que el servicio funcione correctamente, es decir, para realizar un correcto uso del DifferentialDrive (servicio que controla el motor del robot) y del láser.
- **ControlTypes.cs:** Contiene las clases, atributos, contratos y miembros utilizados por el servicio durante su ejecución, así como el estado del servicio.
- **Control.manifest.xml:** Manifiesto del servicio. Contiene los datos fundamentales para lanzar el servicio y que comience a funcionar. Así mismo, lanza los servicios “Entorno” y “Landmarker”, por lo que es éste proyecto el que hay que ejecutar para que la aplicación completa funcione.

Además, estos servicios basan su comportamiento en el entorno generado por el servicio *Entorno*, el cual ya existía, aunque se ha modificado ligeramente para adaptarlo en la medida de lo posible al proyecto que se iba a construir. El servicio *Entorno* se compone de los siguientes elementos:

- **Entorno.cs:** Contiene las principales operaciones para la generación del entorno en el que se va a mover el robot, para la generación del mismo robot, y para su equipamiento (láser, motor, cámara).

Proyecto Fin de Carrera.

Sistema de atención visual para la detección de puntos topológicos de referencia.

- EntornoTypes: Contiene las clases, atributos, contratos y miembros utilizados por el servicio durante su ejecución, así como el estado del servicio.
- Entorno.manifest.xml: Manifiesto del servicio. Contiene los datos fundamentales para lanzar el servicio y que comience a funcionar.

A continuación se expondrán los métodos más importantes utilizados en el servicio de marcado (*Landmarker*) y en el servicio de control:

5.4.1. Servicio *Landmarker*

- **Métodos generales:**
 - **LandmarkerService:** Constructor de la clase.
 - **Start:** Es el método que inicia el servicio y que por tanto realiza las funciones principales del mismo. En este caso se ocupa de activar los principales puertos de comunicación, para a continuación realizar de forma periódica la llamada al método GetFrame, que realizará las funciones principales del servicio.
 - **SubscribeHandler:** Gestiona todas las peticiones de suscripción, por parte de otro servicio, que recibe el servicio de marcado del terreno.
 - **GetFrame:** Como ya se ha indicado, es un método que periódicamente solicita una imagen a la cámara que tiene el robot para trabajar sobre ella. Se ocupa de que no exista ningún error al solicitar la imagen, la convierte a un formato que se pueda manipular fácilmente, y realiza la llamada al método ProcesarImagen() que se encargará de realizar las operaciones necesarias en función de la técnica de análisis de imagen, de las 4 que se han definido anteriormente, que se vaya a emplear. Finalmente, activa el

Proyecto Fin de Carrera.

Sistema de atención visual para la detección de puntos topológicos de referencia.

iterador que permitirá que se ejecute de nuevo el método GetFrame cuando pase el tiempo definido entre cada captación de la imagen.

- **ProcesarImagen:** Realiza las llamadas a los métodos utilizados en función de la técnica de procesamiento de imagen que se esté utilizando. Una vez que se obtiene la matriz binaria con las marcas detectadas, se elimina el ruido de la matriz y se realiza una llamada al método RellenarMarca() para encapsular los datos que se han de mandar como salida a todo servicio que se haya suscrito al servicio de marcado. Finalmente, se envía (si procede) una notificación a todos los servicios suscritos con el mensaje encapsulado, que contiene la imagen original y la matriz binaria con las marcas detectadas.
- **RellenarMarca:** Almacena, en una clase específica, los datos de salida que se envían a cualquier servicio que esté suscrito al servicio de marcado.
- **EliminarRuido:** Método que permite la eliminación de las regiones marcadas de forma individual, es decir, aquéllas que no presentan regiones vecinas también marcadas, tal y como se definió en el apartado de diseño.
- **ActualizarMapa:** Método que actualiza la imagen de la pantalla que muestra las imágenes recibidas por el robot.
- **Métodos relacionados con la técnica de detección con cruces por 128:**
 - **ConvertirImagenActualCruces:** Permite almacenar en una matriz la información disponible en la imagen actual en lo que se refiere a número de cruces por 128 existentes en cada una de las regiones de la imagen reducida. Para ello implementa el algoritmo explicado en el apartado de desarrollo.

Proyecto Fin de Carrera.

Sistema de atención visual para la detección de puntos topológicos de referencia.

- **AnyadirMediaRedCruces:** Utilizado en las primeras n imágenes, permite almacenar en una matriz la información del número medio de cruces por 128 existentes en cada una de las regiones en las que se divide la imagen reducida.
- **GetDiferentesCruces:** Realiza la comparación entre la información de la imagen actual y la información media almacenada anteriormente en lo que se refiere a cruces por cero, para decidir cuáles son las regiones que se han de marcar, y tiene como salida la matriz que representa la imagen binaria.
- **Métodos relacionados con la técnica de detección con histogramas:**
 - **HistogramaActual:** Permite obtener el histograma de la imagen actual para cada una de las regiones en que se divide la imagen reducida, y almacenar toda la información en una matriz. Para ello implementa el algoritmo definido en el apartado de diseño.
 - **AnyadirMediaRedHisto:** Utilizado en las primeras n imágenes, permite almacenar la información de la media de los histogramas de cada una de las regiones de la imagen reducida.
 - **GetDiferentesHisto:** Realiza la comparación entre la información de los histogramas de la imagen actual y la información de la media de los histogramas, almacenada anteriormente, para determinar qué regiones son marcadas. La salida será la matriz que represente la imagen binaria.
- **Métodos relacionados con la técnica de detección con media de colores:**
 - **ConvertirImagenActual:** Convierte la imagen recibida por el robot a una imagen reducida en la que se almacena el valor RGB medio de cada una de las regiones en las que se divide dicha imagen reducida.

Proyecto Fin de Carrera.

Sistema de atención visual para la detección de puntos topológicos de referencia.

- **AnyadirMediaRed:** Durante las primeras n imágenes recibidas, permite realizar una media de la información recibida en las imágenes, y almacenarla en una matriz.
- **GetDiferentes:** Permite realizar la comparación entre la información de la imagen recibida y la almacenada como media anteriormente, en lo que se refiere al color medio de cada una de las regiones en las que se divide la imagen reducida, para conseguir como salida la matriz que representa la imagen binaria con las marcas detectadas.
- **Métodos relacionados con la técnica de detección con media de colores:**
 - **ConvertirImagenActual:** Definido anteriormente.
 - **AnyadirMediaRed:** Definido anteriormente.
 - **ObtenerEstadísticas:** Método que permite, a raíz de la información recabada durante las primeras n imágenes recibidas por el robot, determinar cuáles, de 27 colores básicos, son los colores más frecuentes que se producen en el entorno del robot. Para conseguir esto, se determinará, para cada píxel representativo de cada región de la imagen reducida, cuál es el color al que más se aproxima, analizando sus componentes RGB. Una vez hecho esto, se almacenará cuál es el porcentaje de aparición de cada uno de los 27 colores básicos considerados, en el entorno del robot.
 - **GetDiferentesColores:** Método que permite realizar la comprobación de si el color del píxel representativo de cada una de las regiones en las que se divide la imagen reducida está dentro de los colores más frecuentes del entorno, para en caso contrario marcar la región. La salida de este método, por tanto, será una matriz que represente la imagen binaria.

5.4.2. Servicio de control

- **Métodos generales:**
 - **ControlService:** Constructor de la clase.
 - **Start:** Es el método que inicia el servicio y que por tanto realiza las funciones principales del mismo. En este caso se ocupa de activar los principales puertos de comunicación, para a continuación realizar las suscripciones a los servicios de láser, de motor del robot (*DifferentialDrive*), y al servicio de marcado (*Landmarking*).
- **Métodos relacionados con el láser:**
 - **LaserReplaceNotification:** Método que capta cada notificación del láser procedente del entorno.
 - **LaserRangeFinderUpdateHandler:** Método que recibe la actualización del láser captada por el método anterior, y la trata para obtener y almacenar en el estado general la distancia más cercana a un obstáculo en cada momento.
 - **FindNearestObstacleInCorridor:** Método que utiliza los datos recibidos del láser para calcular la distancia más cercana a un obstáculo.
- **Métodos relacionados con el movimiento del robot:**
 - **DriveUpdateNotification:** Método que capta cada notificación enviada por el motor del robot (*DifferentialDrive*) y que activa el controlador de estado de dicho motor.
 - **DriveUpdateHandler:** Método que obtiene la notificación del motor del robot y llama al método que utiliza los datos más importantes para obtener posición y orientación. Además, llama al método *Moverse()*, que

Proyecto Fin de Carrera.

Sistema de atención visual para la detección de puntos topológicos de referencia.

en función de la información del robot, decidirá cómo ha de moverse el mismo.

- **DriveStateUpdate:** Método que obtiene los datos de posición y orientación del robot y los almacena en el estado del servicio.
- **Moverse:** Método que, en función del estado en el que se encuentra el robot en cada momento, decide qué tipo de movimiento ha de realizar dicho robot (girar a la izquierda, girar a la derecha, avanzar o permanecer parado). Los posibles estados que tiene el robot están definidos en *ControlTypes.cs* y son: CaptarImágenes (al inicio del servicio, cuando el robot únicamente capta la información de su entorno), Moviendo (el robot avanza en línea recta), Parado (el robot no realiza ningún movimiento), Girando (el robot gira hacia un lado u otro en función del sentido de giro que indique el estado), Evitando (el robot gira 150 grados una vez almacenada una marca para no volverla a detectar inmediatamente) y Localizado (el robot ha detectado una marca y ha de dirigirse hacia ella). Igualmente, en el estado Localizado, existe un subestado que indica qué movimiento ha de hacer el robot: GiroIzquierda, si ha de girar a la izquierda, GiroDerecha si ha de hacerlo a la derecha, Avanzar, si ha de seguir en línea recta, Retroceder, si tiene que volver hacia atrás, Parar si se tiene que detener y Finalizado si ya se ha llegado al punto de tomar la imagen y almacenar la posición de la marca.
- **MoveForward:** Método que envía una notificación al motor para que avance en línea recta con una determinada velocidad.
- **Turn:** Método que envía una notificación al motor para que gire con una determinada velocidad.

Proyecto Fin de Carrera.
Sistema de atención visual para la detección de puntos topológicos de referencia.

- **Métodos relacionados con la fase de localización:**
 - **MarcaEncontradaHandler:** Método que se activa cada vez que el servicio de control recibe un mensaje del servicio de marcado. Se encarga de convertir la matriz binaria a un formato más entendible, para a continuación realizar el análisis de dicha imagen binaria. Si existen marcas, se llamará al método DeterminarEstadoLocal() para indicar al robot hacia dónde tiene que moverse para acercarse lo máximo posible a la marca. Si se consigue llegar a la distancia definida para almacenar la imagen, se comprobará si la marca está ya almacenada, y en caso de que no lo esté se almacenará la imagen de la nueva marca.
 - **HayMarcas:** Método que permite determinar si, una vez eliminado el ruido, existen marcas en la imagen binaria.
 - **ObtenerMarcas:** Método que permite obtener todos los ítems o componentes conectadas presentes en la imagen binaria, utilizando el algoritmo descrito en el apartado de diseño. La salida de este método es una lista de listas, donde cada lista contiene todas las coordenadas de cada uno de los diferentes ítems presentes en la imagen binaria.
 - **DeterminarEstadoLocal:** Método que determinará, en caso de existir varias marcas, hacia cuál se debe dirigir el robot, determinando el subestado, dentro del estado “Localizado”, al cuál ha de transitar el robot, dependiendo de la acción que correspondería hacer en ese momento (girar a la derecha, a la izquierda, avanzar, retroceder, parar, o considerar como finalizado el movimiento y almacenar la imagen y la posición de la marca).
 - **GenerarMarca:** Método que genera la posición de la marca a partir del algoritmo definido en el apartado de diseño.

Proyecto Fin de Carrera.
Sistema de atención visual para la detección de puntos topológicos de referencia.

- **AnyadirMarca:** Método que permite añadir la marca generada a la lista que representa la salida global del servicio, así como generar la imagen que representa la marca y su posición, siempre teniendo en cuenta que no haya sido almacenada previamente.
- **CrearImagen:** Método que escribe las coordenadas de una marca sobre la imagen que la representa, y guarda la imagen resultante en el archivo indicado.

5.4.3. Ficheros externos

Existen dos tipos de ficheros externos en este proyecto; el primero de ellos consiste en una librería DLL utilizada para el tratamiento de la imagen. Sin embargo, aunque es necesaria para el funcionamiento de la aplicación, en realidad la librería de tratamiento de imagen se ha utilizado muy poco, por lo que únicamente hay que decir que se escogió dicha librería por el hecho de ser de acceso libre (no comercial) e integrarse fácilmente con MSRDS. El nombre de la librería es AForge.NET, y se puede descargar fácilmente de Internet. En el caso de este proyecto, fue necesario incluir las referencias a dicha librería externa en el servicio *Landmarker*, en concreto a los ficheros “AForge.dll” y “AForge.Math.dll”.

El segundo fichero externo que interesa conocer es un fichero de configuración denominado “Landmarker.xml”, el cuál es necesario incluir en la carpeta “Store”, dentro de la ruta de instalación de MSRDS. Este fichero contiene los datos necesarios para cargar un estado inicial en el servicio de marcado del terreno, con datos que pueden ser configurados desde el inicio. El aspecto de dicho fichero es el siguiente:

```
<?xml version="1.0" encoding="utf-8"?>
<LandmarkerState xmlns:s="http://www.w3.org/2003/05/soap-envelope"
  xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
  xmlns:d="http://schemas.microsoft.com/xw/2004/10/dssp.html"
  xmlns="http://schemas.tempuri.org/2009/06/landmarker.html">
  <VentanaHorizontal>16</VentanaHorizontal>
```

Proyecto Fin de Carrera.
Sistema de atención visual para la detección de puntos topológicos de referencia.

```
<VentanaVertical>12</VentanaVertical>  
<IntervaloCaptura>200</IntervaloCaptura>  
<MaxRecibidos>20</MaxRecibidos>  
</LandmarkerState>
```

Las dos primeras etiquetas son de configuración del archivo, y a partir de ahí se definen los atributos del estado que se quieren configurar externamente. En este caso podemos configurar tanto el tamaño en píxeles de cada una de las regiones en las que se va a dividir la imagen original a la hora de procesarla, el intervalo de tiempo que ha de transcurrir entre una dos peticiones de imagen al servicio de webcam por parte del servicio de marcado, y el número de imágenes recibidas que se va a esperar antes de comenzar a buscar marcas en la imagen, es decir, el número de imágenes que va a componer el banco de datos sobre el que posteriormente se realizarán las comparaciones para decidir si una región de la imagen llama la atención del robot.

Como todo fichero de configuración, el archivo “Landmarker.xml” puede no existir; en tal caso, la aplicación está configurada para que cargue los valores por defecto de los atributos expuestos anteriormente, que son precisamente los que se han mostrado (VentanaHorizontal = 16, VentanaVertical = 12, IntervaloCaptura = 200 y MaxRecibidos = 20).

Asimismo, es necesario disponer de un fichero de configuración que se utiliza para el correcto funcionamiento del servicio de Entorno. Dicho fichero se denomina “mazesimuladorra.config.xml” y no ha sido desarrollado para este proyecto en concreto.

5.5. Manual de usuario

El manual de usuario de la aplicación es bastante sencillo de describir. Únicamente se basa en tener instalada la última versión de Microsoft Robotics Developer Studio (MSRDS), y la carpeta de instalación incluir la carpeta “PFC” que se proporciona como material de este proyecto.

Dentro de dicha carpeta “PFC”, deben existir los siguientes elementos:

Proyecto Fin de Carrera.
Sistema de atención visual para la detección de puntos topológicos de referencia.

- Carpeta Config: Contiene los ficheros para la configuración inicial del entorno.
 - Mazesimulorra.config.xml: Fichero de configuración del entorno.
- Carpeta Control: Contiene los ficheros pertenecientes al servicio “Control”.
- Carpeta Entorno: Contiene los ficheros pertenecientes al servicio “Entorno”.
- Carpeta Simulación: Contiene los archivos de imagen que representan el entorno.
 - Entorno.bmp: Fichero con la imagen en dos dimensiones para construir el entorno en tres dimensiones.
- Carpeta Landmarker: Contiene los ficheros pertenecientes al servicio “Landmarker”.
- Carpeta Resultados: Carpeta en la que se guardarán las imágenes resultantes de ejecutar el programa, con los nombres “MarcaX.bmp”, donde X es un número consecutivo comenzando por el 1, en cada ejecución.
- Carpeta Externos: Carpeta en la que se almacenan los ficheros externos necesarios para ejecutar la aplicación.
 - AForge.dll: Librería que será necesario incluir como referencia en el servicio *Landmarker* (o, si ya está incluida, comprobar que la ruta del archivo es correcta).
 - AForge.Imaging.dll: Librería que será necesario incluir como referencia en el servicio *Landmarker* (o, si ya está incluida, comprobar que la ruta del archivo es correcta).
 - Landmarker.xml: Fichero de configuración del servicio *Landmarker*, será necesario copiarla a la carpeta “Store”, en el directorio raíz del programa Microsoft Robotics Developer Studio.

Proyecto Fin de Carrera.
Sistema de atención visual para la detección de puntos topológicos de referencia.

Por lo expuesto anteriormente, es necesario asegurarse de que el nombre de la imagen que representa al entorno en dos dimensiones sea Entorno.bmp, y se encuentre dentro de la carpeta “PFC/Simulación”, ya que en caso contrario el programa no cargará el entorno.

Una vez que se tienen dispuestos todos los elementos únicamente hay que ejecutar, desde el programa Visual Studio 2008, el servicio “Control”, pulsando F5 o haciendo clic sobre el icono de “Play”, en la parte superior de la pantalla. Otra forma de ejecutarlo es a través del “*Command Prompt*” o pantalla de comandos del programa MSRDS, mediante la siguiente instrucción:

>> dsshost /port:50000 /manifest: “PFC/Control/Control.manifest.xml”

6. Experimentación y resultados

6.1. Descripción del entorno de pruebas

El entorno de pruebas utilizado se conseguirá gracias al simulador de Microsoft Robotics Developer Studio, y sobre todo gracias al servicio denominado *Entorno*, ya nombrado anteriormente, que posee una serie de métodos que permiten tomar una imagen en dos dimensiones y, a partir de ella, ir generando todas las entidades dispuestas en la misma en tres dimensiones. Únicamente será necesario definir unos colores específicos en la imagen que se dibuja, y el programa, automáticamente, genera el entorno deseado. Un ejemplo de dicho programa sería el siguiente:

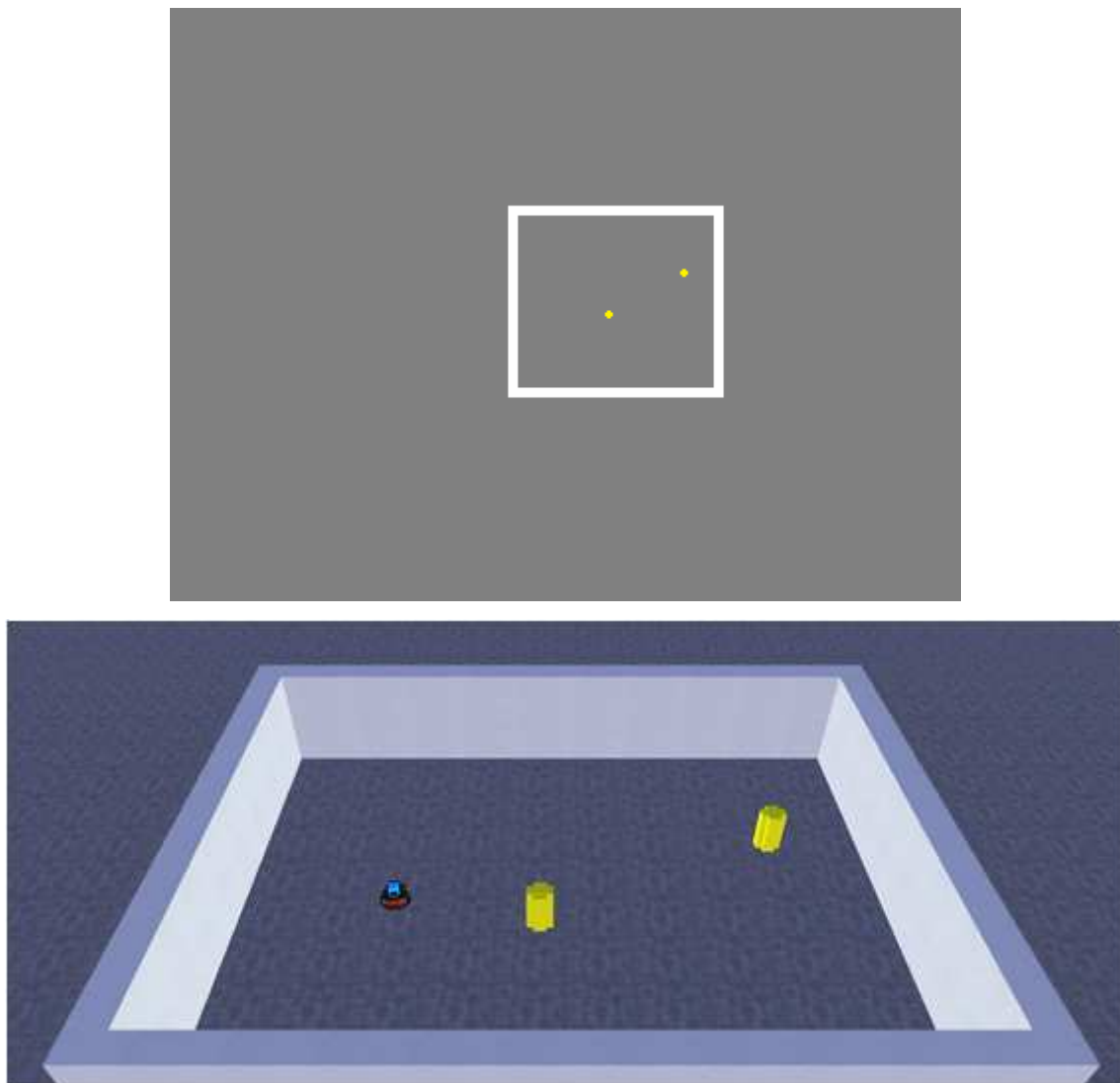


Figura 24: Ejemplo de transformación 2-D a 3-D (1)

Proyecto Fin de Carrera.
Sistema de atención visual para la detección de puntos topológicos de referencia.

Y otro ejemplo:

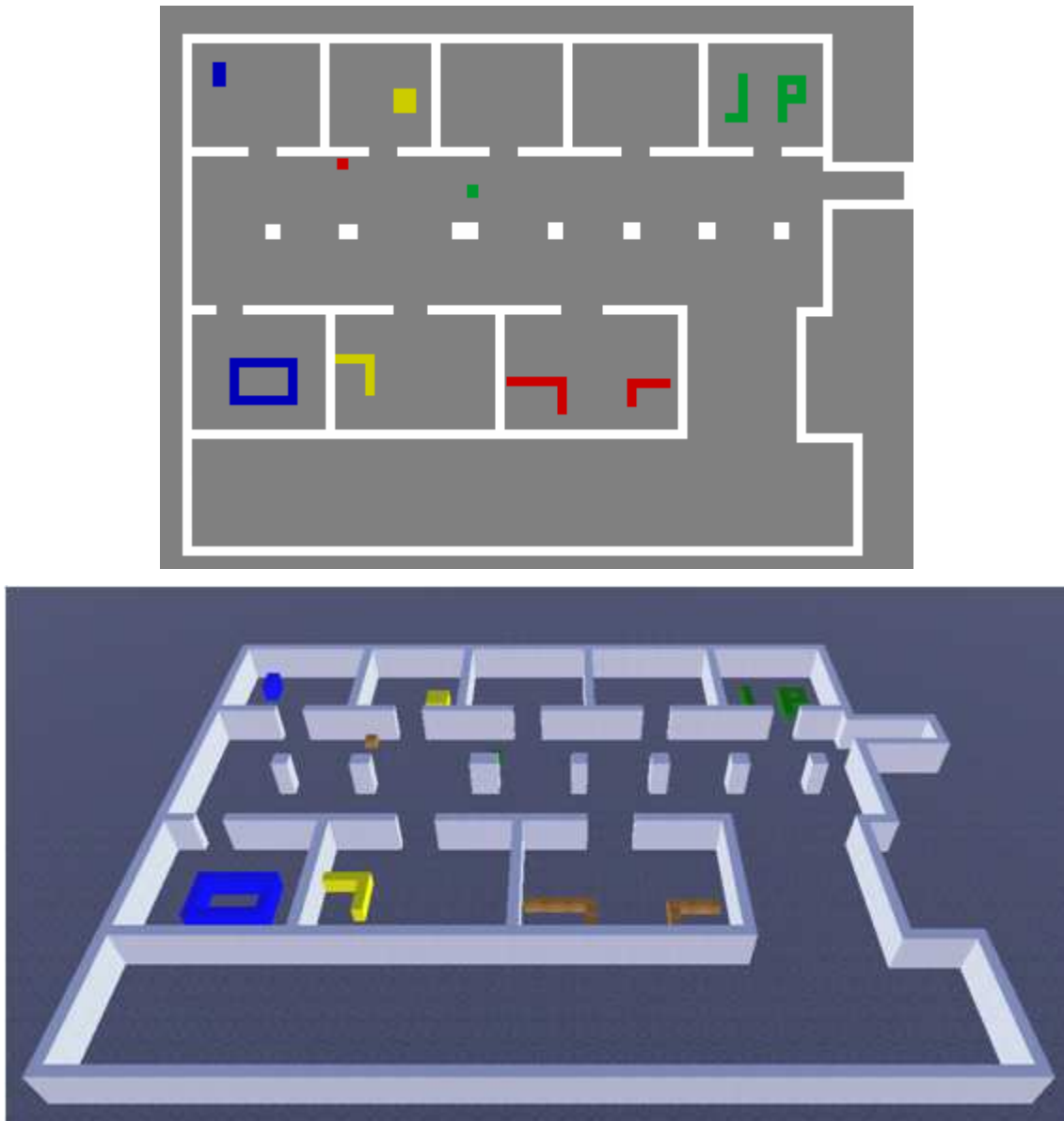


Figura 25: Ejemplo de transformación 2-D a 3-D (2)

Como se puede observar en las imágenes anteriores, el color grisáceo en la imagen en dos dimensiones representa el suelo una vez convertida a tres dimensiones, y el color blanco se interpreta como un muro (de mayor altura que el resto de los colores). Finalmente, cualquier otro color se representará como un elemento de un tamaño más reducido que el muro, y del color especificado (salvo algunos colores, como el rojo que se representa con una textura de madera).

Proyecto Fin de Carrera.

Sistema de atención visual para la detección de puntos topológicos de referencia.

Hay que notar también que por defecto, al ejecutar el programa, el robot Pioneer 3-DX aparece en el centro justo de la imagen en dos dimensiones que se ha utilizado como base, por lo que es importante que al diseñar dicha imagen en 2D se considere esta eventualidad para evitar que el robot quede fuera de los límites de los muros que se coloquen, lo cual, al tener el robot, antes de detectar una marca, una navegación aleatoria, podría significar que se alejara indefinidamente de los límites marcados por los muros.

Además, un dato muy importante sobre la simulación, es que la conversión de dos dimensiones a tres, en lo que se refiere a las medidas del mundo simulado, se realiza de tal manera que un píxel en la imagen de dos dimensiones equivale a una distancia de 10 centímetros en la imagen de tres dimensiones (mundo virtual). Por tanto, esto será muy importante en el momento de determinar si la posición indicada por el robot para una determinada marca es correcta o no.

Se han tenido que realizar ligeras modificaciones para facilitar el funcionamiento del robot. La más importante se basa en la altura de los muros, ya que si se colocaban unos muros tan bajos como dibujaba el simulador, dentro de las imágenes que captaba el robot se incluía el cielo, que se representa con una textura predefinida de color blanco y azul. Esta textura presentaba conflicto prácticamente con todas las técnicas de análisis de la imagen que se han utilizado, por lo que se decidió aumentar la altura de los muros, para que el robot no llegue a obtener imágenes del cielo y se centre en lo que hay dentro de los muros.

Debido a que el robot tiene una navegación automática (se considera un robot autónomo móvil, aunque la navegación sea aleatoria en algunos casos), al ejecutar el programa no aparecerá ninguna pantalla de navegación para que el usuario maneje dicho robot (pantalla denominada *dashboard*), sino que directamente el robot comenzará a moverse. Sin embargo, se ha diseñado otra pantalla para que el usuario, además de visualizar el entorno completo y poder moverse por él, pueda ver cuáles las imágenes que está obteniendo el robot en tiempo real, y por tanto cuáles son las imágenes que el robot está analizando en busca de marcas. Dicha pantalla se denomina *CameraForm*, y si por ejemplo, el robot está en la siguiente posición:

Proyecto Fin de Carrera.
Sistema de atención visual para la detección de puntos topológicos de referencia.

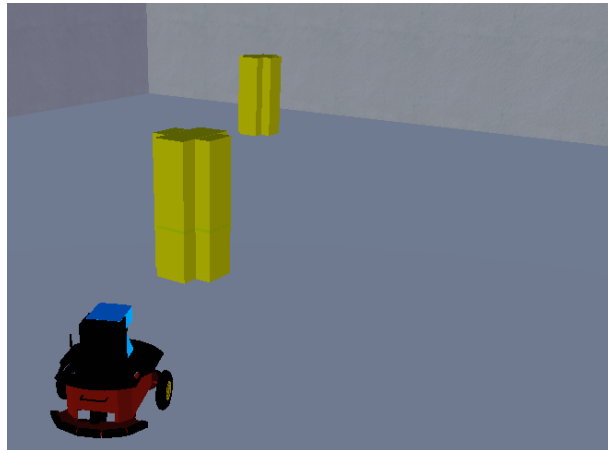


Figura 26: Visión exterior del entorno

La cámara que muestra las imágenes que obtiene el robot, tendrá el siguiente aspecto en este momento determinado:

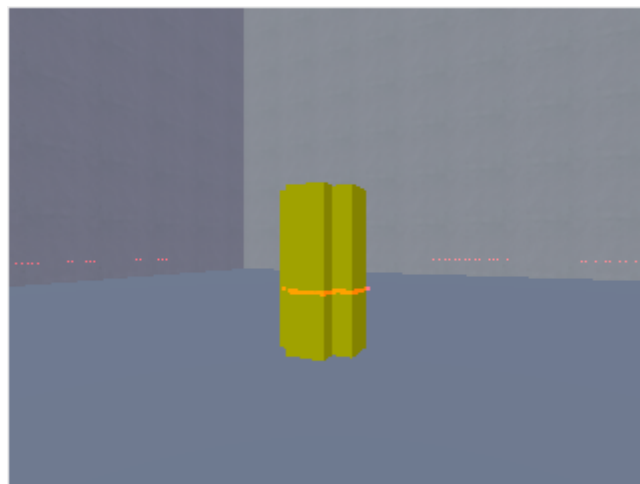


Figura 27: Visión desde el robot del entorno

6.2. Pruebas realizadas

Las pruebas realizadas se han orientado hacia varios objetivos. El primero de ellos ha sido comprobar cuál de las cuatro técnicas de análisis de la imagen utilizadas y descritas en esta memoria funciona mejor. Se ha utilizado el mismo entorno para todas las pruebas, para tener una comparativa fiable entre las 4 técnicas. Las pruebas realizadas en este aspecto han sido las siguientes:

Proyecto Fin de Carrera.
Sistema de atención visual para la detección de puntos topológicos de referencia.

- Prueba 1: Funcionamiento de la técnica “Cruces por 128”.
- Prueba 2: Funcionamiento de la técnica “Histogramas”.
- Prueba 3: Funcionamiento de la técnica “Media de colores” con colores homogéneos.
- Prueba 4: Funcionamiento de la técnica “Media de colores” con colores no homogéneos.
- Prueba 5: Funcionamiento de la técnica “Colores más frecuentes” con colores homogéneos.
- Prueba 6: Funcionamiento de la técnica “Colores más frecuentes” con colores no homogéneos.

El segundo aspecto hacia el que se ha dirigido la batería de pruebas ha sido, una vez seleccionada una técnica para el análisis de la imagen, comprobar que los algoritmos utilizados en la fase de localización funcionan correctamente, y por tanto el robot es capaz de detectar las balizas que se han dispuesto dentro de los límites de los muros. Las pruebas realizadas en esta fase son:

- Prueba 7: Detección de marcas en entorno simple.
- Prueba 8: Detección de marcas en entorno complejo.
- Prueba 9: Detección de marcas no homogéneas.

6.3. Resultados obtenidos

6.3.1. Prueba 1 (Cruces por 128)

El entorno utilizado para esta prueba ha sido un rectángulo simple con una única baliza de color amarillo, que el robot ha de detectar. A lo largo de la prueba, se puede observar que detecta la baliza, pero confunde como marcas muchas otras cosas que no lo son, como las paredes o el suelo en muchas ocasiones, tal y como se puede observar en la imagen siguiente:

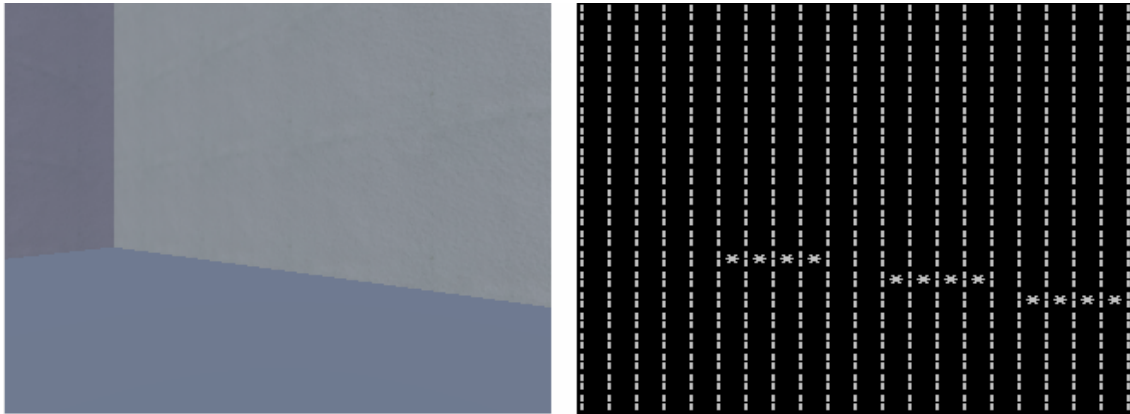


Figura 28: Resultados prueba 1

Como se puede observar, en este caso en el que el robot se está aproximando a una esquina de la pared, es decir, que no está visualizando ninguna marca, interpreta que sí existe dicha marca, ya que está detectando como tal la separación entre la pared y el suelo. Por lo que se puede determinar que esta técnica provoca resultados fallidos incluso en casos de colores homogéneos de paredes y suelo.

La imagen expuesta para ilustrar el funcionamiento de esta técnica es simplemente un ejemplo de lo ocurrido a lo largo de la navegación del robot. Estos fallos que se han comentado (la detección de falsas marcas, fundamentalmente), ocurren en todo momento, por lo que la técnica no es aceptable en ningún caso.

6.3.2. Prueba 2 (Histogramas)

El entorno utilizado para esta prueba ha sido el mismo que el utilizado para la prueba anterior, es decir, un rectángulo simple con una única baliza de color amarillo, que el robot ha de detectar. Los resultados obtenidos son también poco satisfactorios, ya que al igual que ocurría en la prueba número 1, existen momentos en los que el robot determina que son marcas determinadas elementos de la imagen que no lo son. Este comportamiento queda ilustrado en la imagen siguiente:

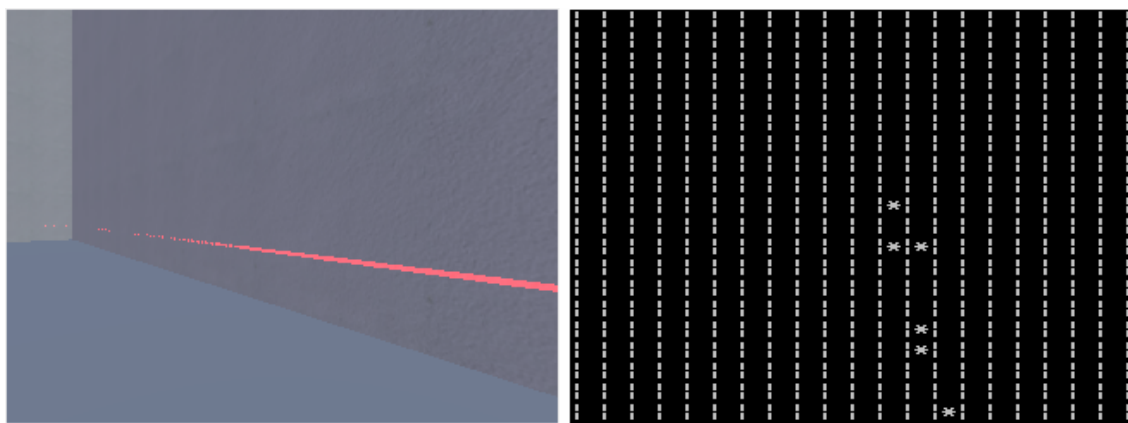


Figura 29: Resultados prueba 2

Como se puede observar en esta imagen, el robot detecta marcas que ni siquiera siguen un esquema lógico (es decir, no coincide con la unión de suelo y paredes), por lo que se puede desestimar también esta técnica por arrojar resultados poco satisfactorios. No están muy claras las razones de que esta técnica presente los comportamientos que se han observado, aunque es posible que los histogramas funcionen mejor para la detección de variaciones en la imagen que para la detección de color, y por eso se marcan zonas que no tienen que ver con colores no predominantes, sino que posiblemente tengan que ver con variaciones en la imagen.

6.3.3. Prueba 3 (Media de colores con colores homogéneos)

Se utilizará el mismo entorno para el robot que en las pruebas anteriores. En esta prueba, se observa que el robot parece detectar mucho mejor las marcas (balizas amarillas), ya que, sobre todo, no detecta otras marcas cuando no existen, como ocurría con las dos técnicas anteriores. Sin embargo, existe un pequeño problema, y es que

Proyecto Fin de Carrera.

Sistema de atención visual para la detección de puntos topológicos de referencia.

necesita estar muy cerca de las marcas para comenzar a detectarlas, tal y como se puede ver en la siguiente imagen.

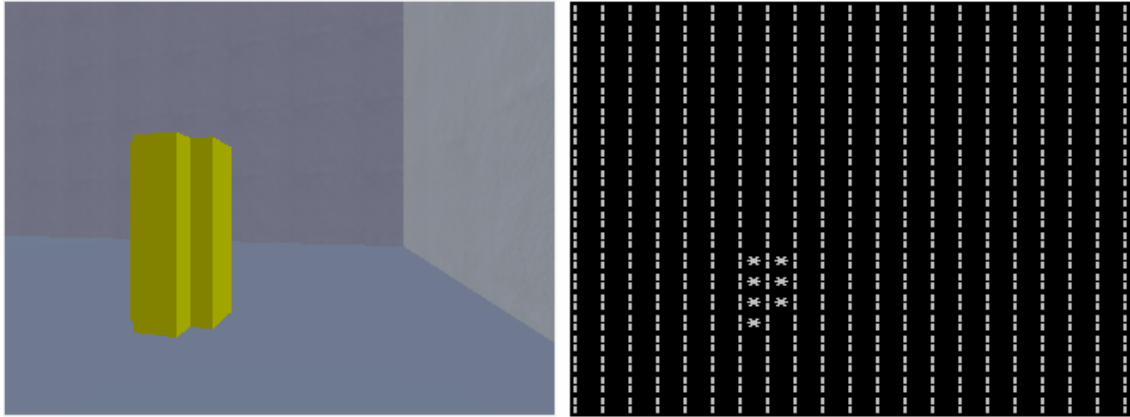


Figura 30: Resultados prueba 3

Sin embargo, como parece que funciona aceptablemente, se va a realizar una prueba con unos colores de paredes y suelo más heterogéneos, para ver si no provocan confusión en el robot.

6.3.4. Prueba 4 (Media de colores con colores no homogéneos)

En esta prueba concreta, se utilizará el mismo entorno de pruebas que el descrito en las pruebas anteriores, pero con una ligera modificación, ya que se va a modificar el color del suelo para que no se parezca tanto al de la pared (ahora, el color de la pared seguirá siendo gris, y el del suelo será verde). Si la técnica fuese robusta en lo que se refiere a la variación de color en el entorno, debería funcionar también. El resultado se puede observar en la siguiente imagen:

Proyecto Fin de Carrera.
Sistema de atención visual para la detección de puntos topológicos de referencia.

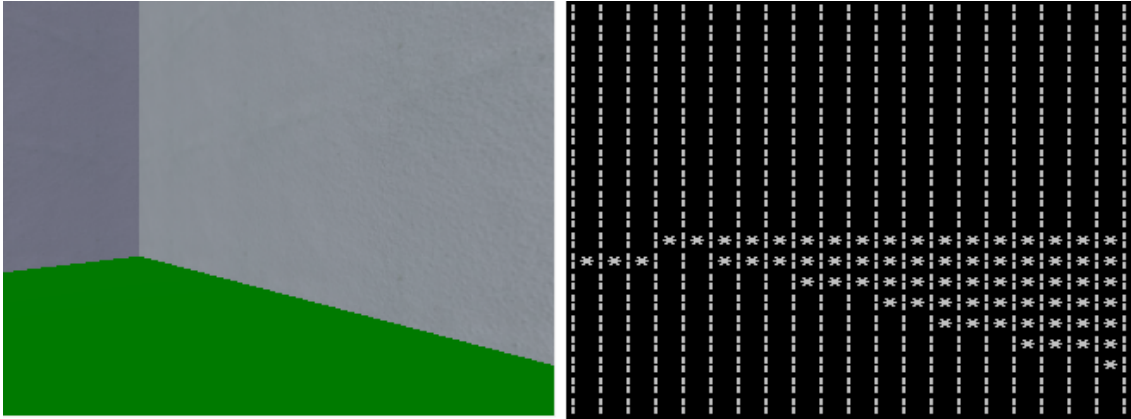


Figura 31: Resultados prueba 4

Como podemos ver, la técnica falla estrepitosamente cuando no tenemos colores homogéneos en paredes y suelo, detectando marcas donde no las hay, concretamente en la unión entre pared y suelo, tal y como ocurría en la prueba 1, lo cual nos indica que es uno de los lugares más conflictivos para conseguir una buena técnica de detección de marcas. A continuación, por tanto, se pasará a probar la técnica de colores más frecuentes, que es más general y no depende tanto del lugar que represente la región de la imagen reducida, sino que se centra en los colores independientemente de sus posiciones.

6.3.5. Prueba 5 (Colores más frecuentes con colores homogéneos)

En este caso, se vuelve a utilizar el entorno de las pruebas 1, 2 y 3, es decir, aquél con colores homogéneos (grisáceos) de paredes y suelo. Los resultados arrojados por esta técnica son totalmente satisfactorios, como se puede ver en la siguiente imagen:

Proyecto Fin de Carrera.
Sistema de atención visual para la detección de puntos topológicos de referencia.

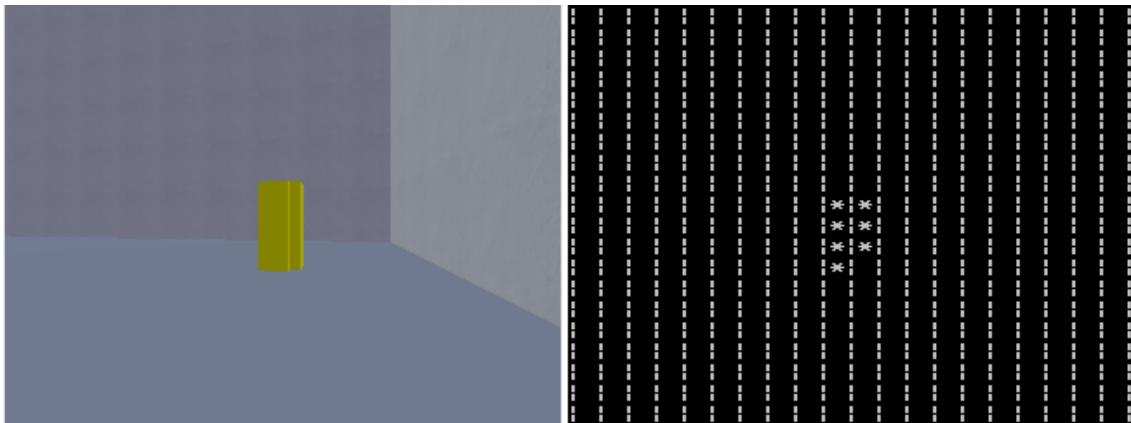


Figura 32: Resultados prueba 5

Al igual que ocurría con la técnica de media de colores, cuando los colores son homogéneos, el robot detecta las marcas correctamente, y además no confunde marcas con otros elementos que no lo son. Además, parece que detecta las marcas desde posiciones más alejadas, lo cual puede ser una buena señal. Por tanto, es momento de comprobar si esta técnica funciona correctamente también con colores no homogéneos.

6.3.6. Prueba 6 (Colores más frecuentes con colores no homogéneos)

Tal y como hicimos en la prueba 4, vamos a probar la técnica utilizando el mismo entorno de prueba, pero variando el color del suelo para que ahora sea verde y no exista mucha igualdad con el color de la pared. Teniendo en cuenta el algoritmo que utiliza esta técnica de colores más frecuentes (explicado en la parte de diseño), la prueba debería ser satisfactoria:

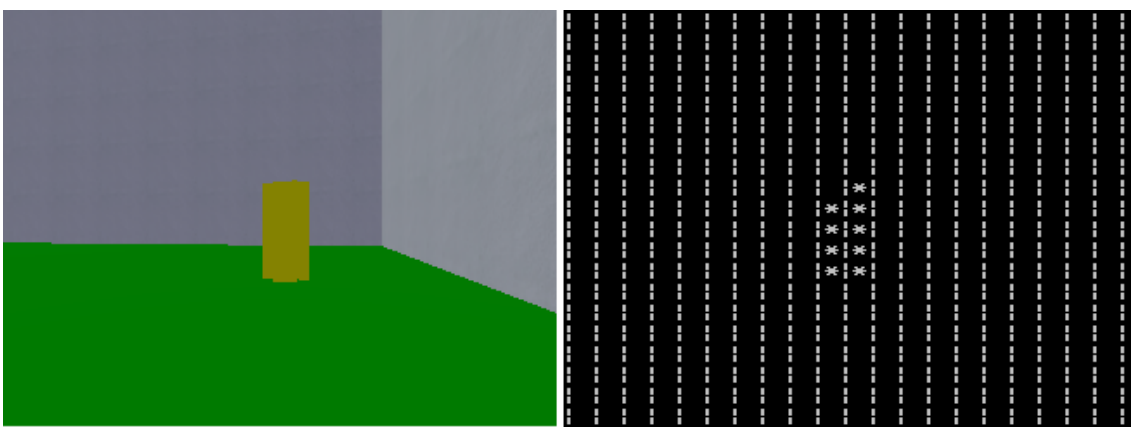


Figura 33: Resultados prueba 6

Proyecto Fin de Carrera.
Sistema de atención visual para la detección de puntos topológicos de referencia.

Como se puede observar, la técnica también detecta bien las marcas del entorno, y no se producen errores por detección de marcas que realmente no lo son, por lo que podemos ver que la técnica de colores más frecuentes es la más robusta de las cuatro que se han implementado.

6.3.7. Prueba 7 (Detección de marcas en entorno simple)

En esta primera prueba de la fase de localización, se persigue comprobar si la implementación realizada, que en principio permite obtener las imágenes de las marcas presentes en el entorno del robot, funciona correctamente. Por ello, para la primera prueba se utilizará un entorno simple con dos balizas amarillas que serán las marcas que el robot tendrá que detectar y localizar. Este entorno se muestra a continuación (Figura 34: Entorno prueba 7):

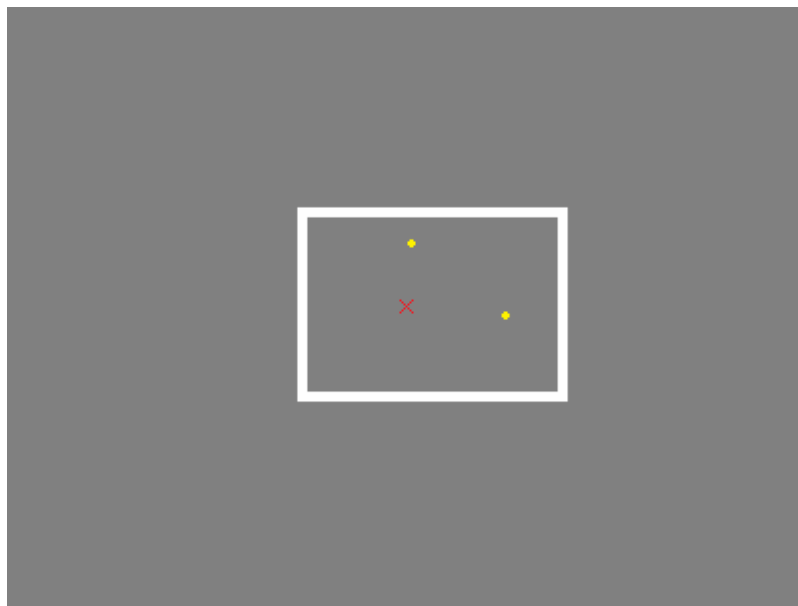


Figura 34: Entorno prueba 7

En la imagen se muestra el recinto en el que se va a encontrar el robot, y las posiciones de las dos balizas amarillas que éste tiene que detectar y localizar. La cruz roja, que indica el centro de la imagen, nos indica también dónde va a comenzar el robot su exploración, y por tanto, cuál es el punto de referencia (0,0) del robot para conseguir las posiciones relativas de las marcas.

Proyecto Fin de Carrera.
Sistema de atención visual para la detección de puntos topológicos de referencia.

Se ejecuta la aplicación hasta que se observa que se han almacenado las posiciones de las dos marcas en dos imágenes diferentes, que se muestran a continuación (Figura 35: Marca 1 de la prueba 7 y Figura 36: Marca 2 de la prueba 7):

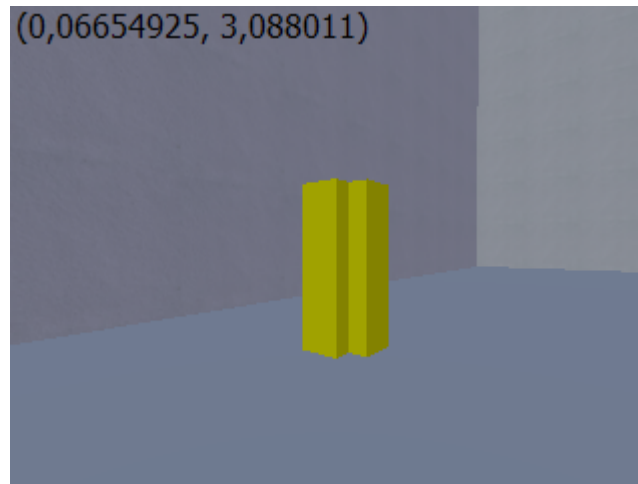


Figura 35: Marca 1 de la prueba 7

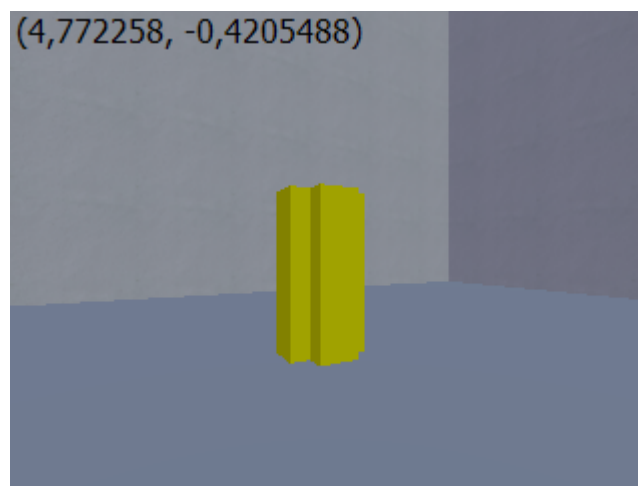


Figura 36: Marca 2 de la prueba 7

Para saber si los resultados han sido satisfactorios, hay que analizar las posiciones que nos indican las coordenadas de las marcas, y contrastarlas con la imagen del entorno. La primera marca indica unas coordenadas, redondeando los números, de (0.07, 3.09). Esto significa que la marca se encuentra a 7 centímetros a la derecha y 3 metros y 9 centímetros por encima de la posición inicial del robot. Traducido a píxeles en la imagen en 2D del entorno, la marca se encontrará aproximadamente un píxel a la derecha y 31 píxeles por encima de la posición inicial del robot. En cuanto a la segunda marca, sus coordenadas son (4.77, -0.42), por lo que la marca ha de estar 4 metros y 77

Proyecto Fin de Carrera.

Sistema de atención visual para la detección de puntos topológicos de referencia.

centímetros a la derecha y 42 centímetros por debajo de la posición inicial del robot. Por tanto, esto se traduce a 48 píxeles a la derecha y 4 píxeles por debajo de la posición del robot. Determinando estas posiciones en la imagen del entorno, tenemos el siguiente resultado:

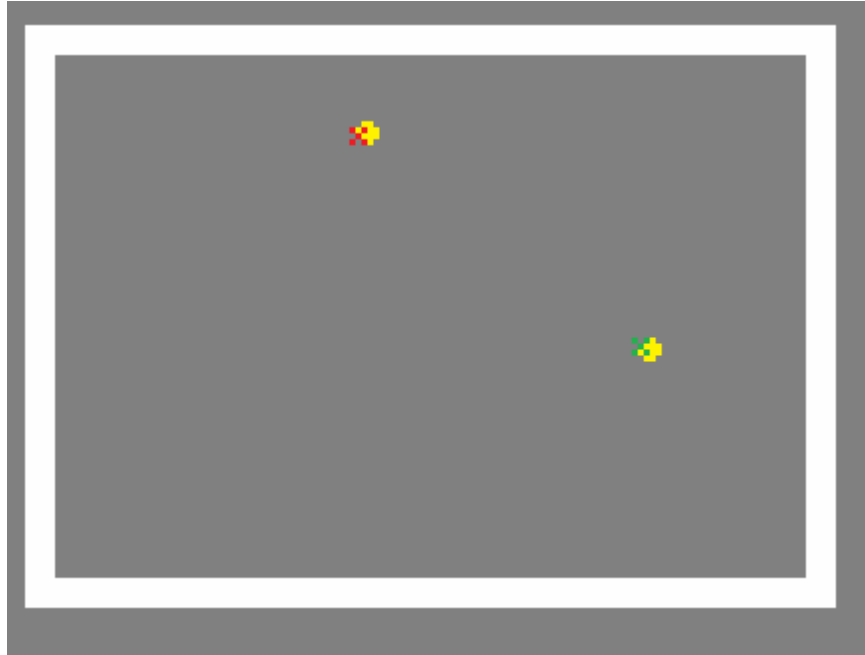


Figura 37: Resultados prueba 7

El centro de la cruz roja de la imagen corresponde a las coordenadas proporcionadas por la aplicación para la primera marca, mientras que el centro de la cruz verde corresponde a las coordenadas proporcionadas para la segunda marca. Como se puede observar, el centro de cada cruz indica un píxel perteneciente a la marca, lo cual refleja que la prueba ha resultado completamente satisfactoria, ya que se han detectado las dos marcas existentes, y no se ha detectado ninguna marca falsa. Lógicamente, las coordenadas que el robot almacene dependerán de la orientación desde la que se acerque a la marca, sin embargo, la posición es correcta, ya que en las coordenadas que el robot indica, existe en efecto una marca, aunque su extensión sea mayor.

En cuanto al rendimiento de la aplicación, el robot ha tardado aproximadamente 4 minutos en encontrar y almacenar las dos marcas. Es un tiempo alto para la navegación por un entorno tan pequeño, pero esto es debido a que la navegación global

Proyecto Fin de Carrera.

Sistema de atención visual para la detección de puntos topológicos de referencia.

es aleatoria, por lo que puede pasar un tiempo hasta que el robot encuentre una marca, cambie su comportamiento a navegación local, y se acerque hasta la marca.

6.3.8. Prueba 8 (Detección de marcas en entorno complejo)

La siguiente prueba de localización consistirá en encontrar marcas homogéneas (balizas amarillas, como las que se han utilizado en la prueba anterior), pero en un entorno un poco más complejo, en el que no exista únicamente una “habitación”, sino que el robot tenga que explorar un poco más para encontrar las marcas. El entorno elegido se ilustra en la imagen siguiente (Figura 38: Entorno prueba 8):

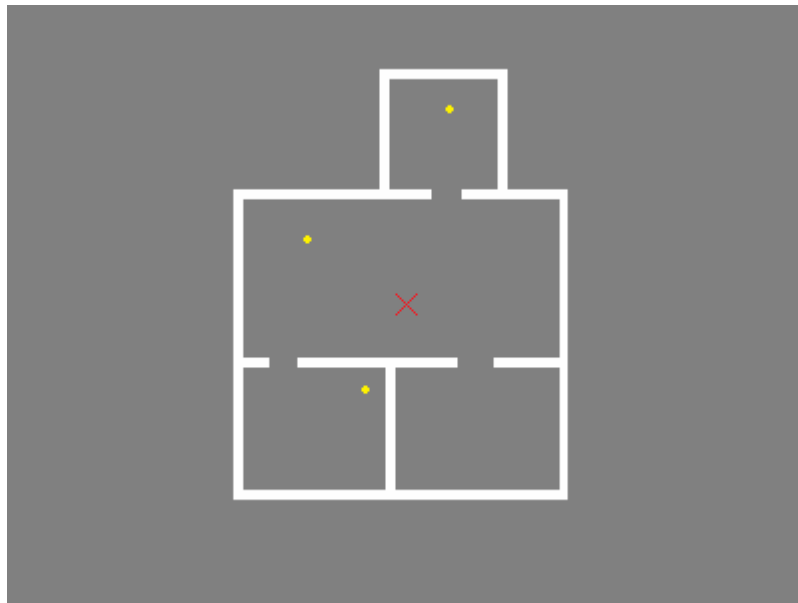


Figura 38: Entorno prueba 8

En este entorno, como se observa, se ha utilizado una disposición de muros más complicada, como la que podría representar un mapa de una casa o de un edificio con habitaciones o despachos. Al igual que en la prueba anterior, la cruz nos indica la posición inicial (0,0) del robot, a partir de la cuál éste determinará las posiciones relativas de las marcas que encuentre.

Las imágenes que ofrece el robot tras realizar el análisis de su entorno son las siguientes:

Proyecto Fin de Carrera.
Sistema de atención visual para la detección de puntos topológicos de referencia.

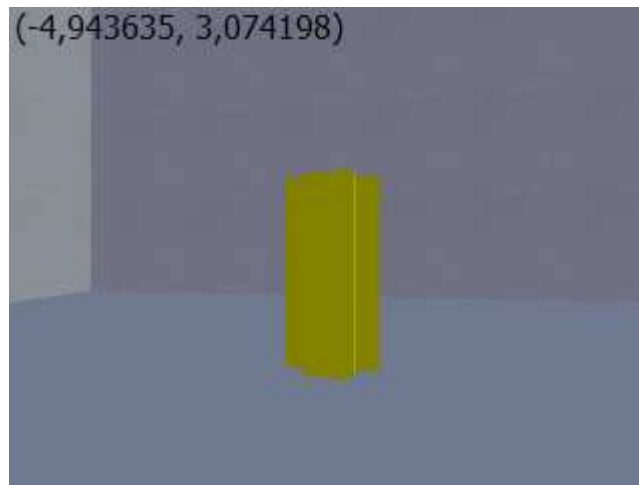


Figura 39: Marca 1 de la prueba 8

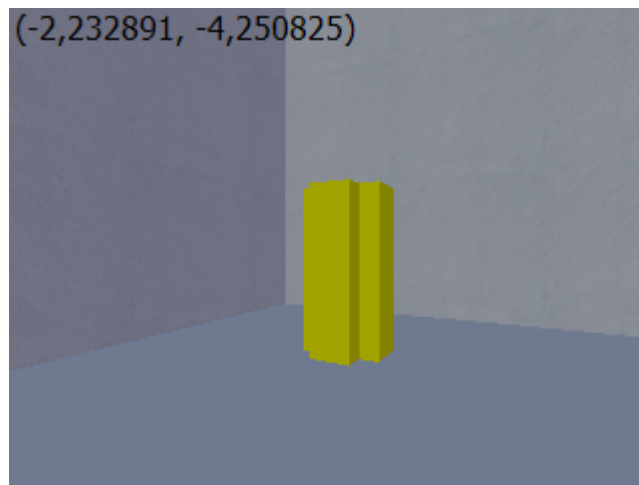


Figura 40: Marca 2 de la prueba 8

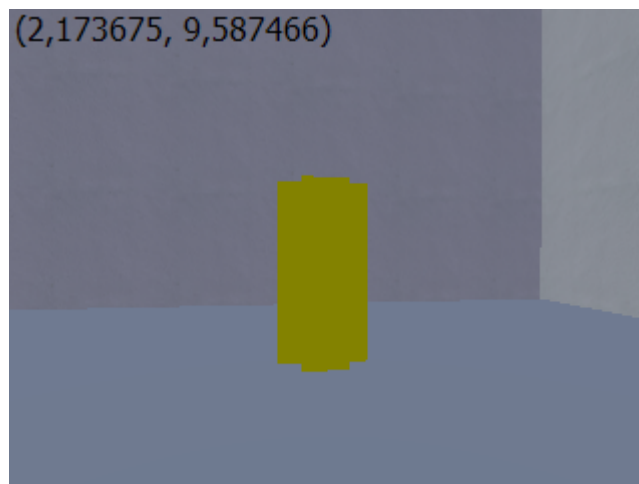


Figura 41: Marca 3 de la prueba 8

Proyecto Fin de Carrera.
Sistema de atención visual para la detección de puntos topológicos de referencia.

Y, realizando el mismo análisis que en la prueba número 7, tenemos que la primera marca (-4.94, 3.07) ha de estar 49 píxeles a la izquierda y 31 píxeles por encima de la posición inicial del robot, la segunda marca (-2.23, -4.25) ha de estar 22 píxeles a la izquierda y 43 por debajo de la posición inicial, y la tercera marca (2.17, 9.59) ha de estar 22 píxeles a la derecha y 96 píxeles por encima de la posición inicial.

Comprobando estos datos, obtenemos el siguiente resultado:

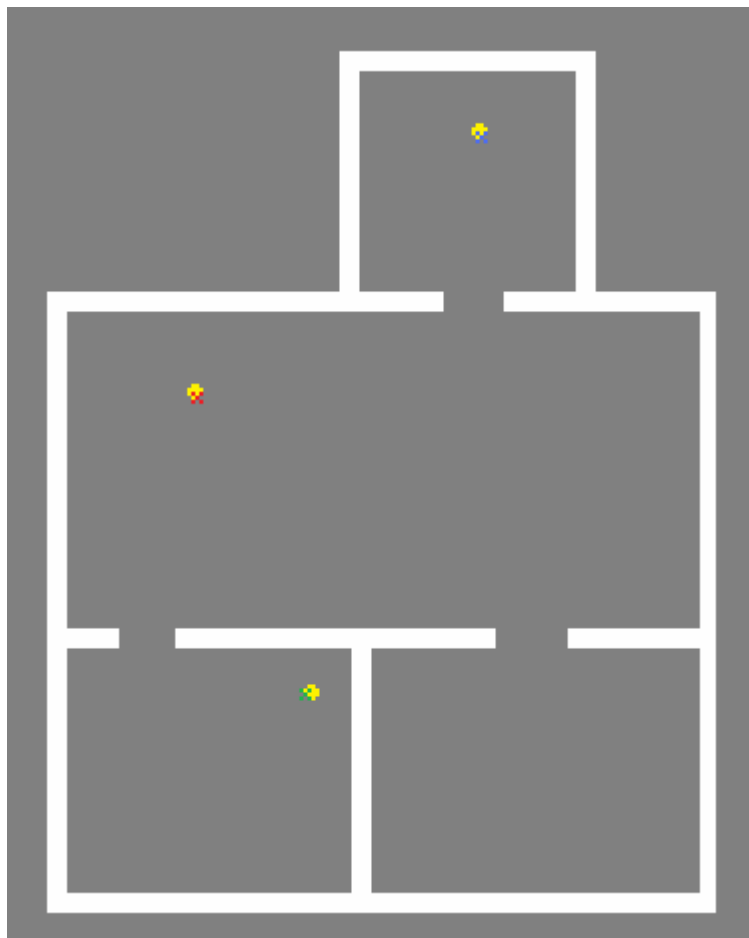


Figura 42: Resultados prueba 7

Como se puede observar, las tres marcas (marca 1 en rojo, marca 2 en verde y marca 3 en azul) coinciden con la posición de las balizas en el mapa inicial, por lo que se puede decir que los resultados de esta prueba son satisfactorios. Sin embargo, para conseguir comprobar si el robot localizaba correctamente las marcas, ha sido necesario parar la ejecución del programa en algunas ocasiones y colocar al robot en determinadas posiciones más favorables para que las marcas entrasen fácilmente en su campo de visión, ya que debido a la navegación global aleatoria, es complicado que el robot

Proyecto Fin de Carrera.

Sistema de atención visual para la detección de puntos topológicos de referencia.

explore en un tiempo razonable todo el entorno para encontrar las marcas. Por ello, queda bastante comprobado que la navegación aleatoria no es óptima de ninguna manera, por lo que sería interesante, en futuros proyectos, implementar otro tipo de navegación que mejore el sistema. Ésta y otras consideraciones se comentarán en el capítulo dedicado a trabajos futuros (Capítulo 8: Trabajos futuros).

6.3.9. Prueba 9 (Detección de marcas no homogéneas)

La prueba número 9 se basa en comprobar si el sistema desarrollado detecta y localiza correctamente otros tipos de marcas diferentes a las utilizadas hasta ahora. Para ello, además de una baliza amarilla se van a utilizar dos balizas más de formas y colores diferentes (una verde y otra con textura de madera). Sin embargo, para evitar que el robot invierta demasiado tiempo navegando por el entorno, éste será bastante más sencillo que en la prueba número 8. Por tanto, el mapa en dos dimensiones del entorno es el siguiente:

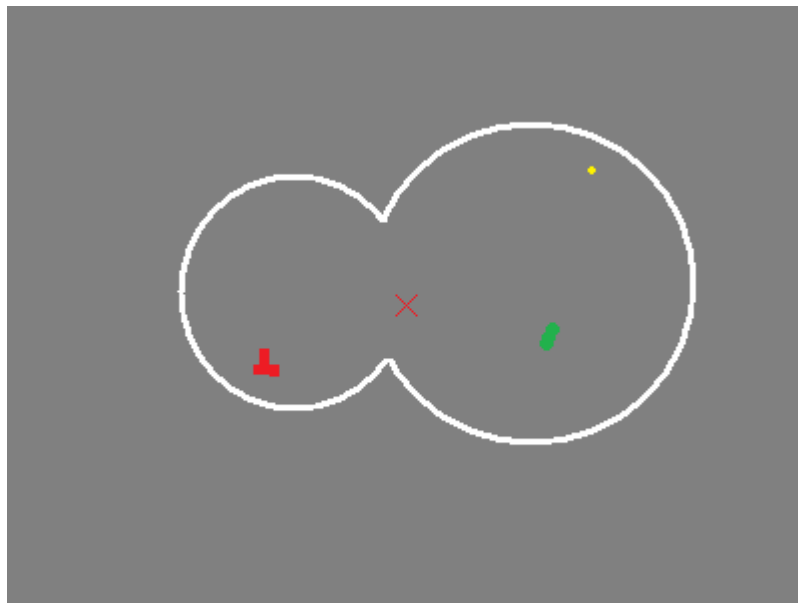


Figura 43: Entorno prueba 9

Para modificar en algún aspecto los entornos utilizados, se ha optado por utilizar un entorno con una forma redonda, para comprobar que el robot navega correctamente también cuando se encuentra con paredes curvadas. Además, tal y como se puede ver en la imagen, ahora tenemos tres marcas diferentes: una baliza amarilla, igual que las que

Proyecto Fin de Carrera.

Sistema de atención visual para la detección de puntos topológicos de referencia.

se han utilizado hasta ahora, un módulo con formas rectas con textura de madera, representado en la imagen con color rojo, y un módulo con formas más o menos curvadas de color verde. Igual que en pruebas, anteriores, la cruz de color rojo indica el punto en el que el robot comienza su navegación (0,0).

Las imágenes obtenidas como marcas de referencia una vez ejecutado el programa son las siguientes:

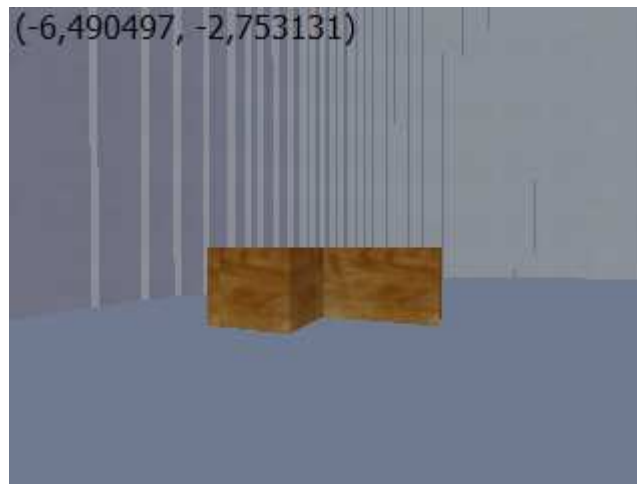


Figura 44: Marca 1 de la prueba 9

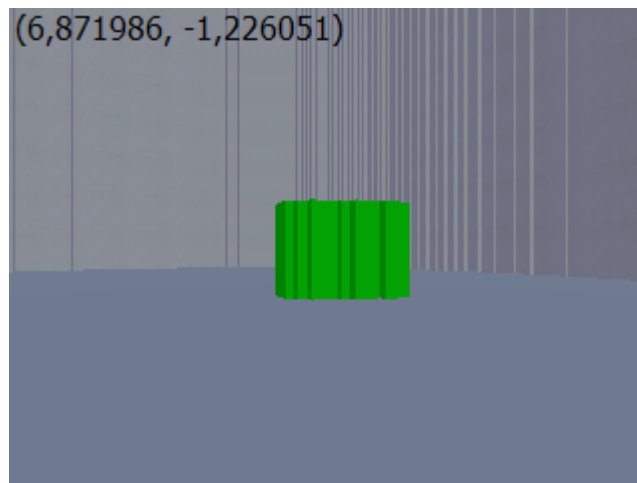


Figura 45: Marca 2 de la prueba 9

Proyecto Fin de Carrera.
Sistema de atención visual para la detección de puntos topológicos de referencia.

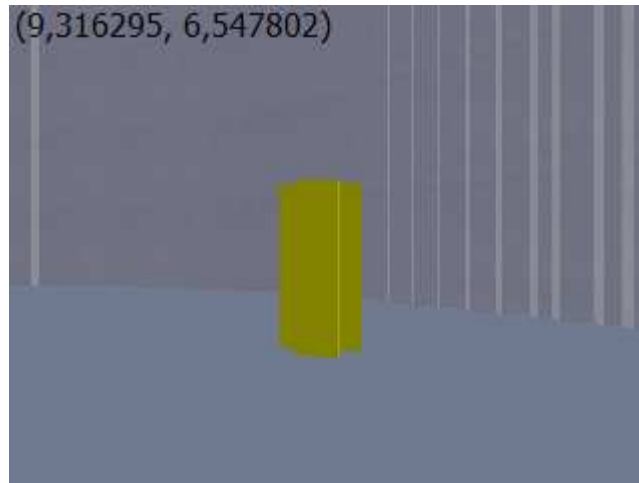


Figura 46: Marca 3 de la prueba 9

Al igual que se ha hecho en las dos pruebas anteriores, utilizaremos las posiciones sobrescritas en las marcas para comprobar si la prueba ha tenido éxito. La primera marca $(-6.49, -2.75)$ ha de estar 65 píxeles a la izquierda y 28 píxeles por debajo de la posición original del robot, en la imagen en dos dimensiones. La segunda marca $(6.87, -1.23)$ ha de estar 69 píxeles a la derecha y 12 píxeles por debajo de la posición $(0,0)$. Por último, la tercer marca $(9.32, 6.55)$ ha de estar 93 píxeles a la derecha y 66 píxeles por encima de la posición original del robot.

Comprobamos dichos datos en la imagen del entorno para verificar si los resultados son satisfactorios:

Proyecto Fin de Carrera.
Sistema de atención visual para la detección de puntos topológicos de referencia.

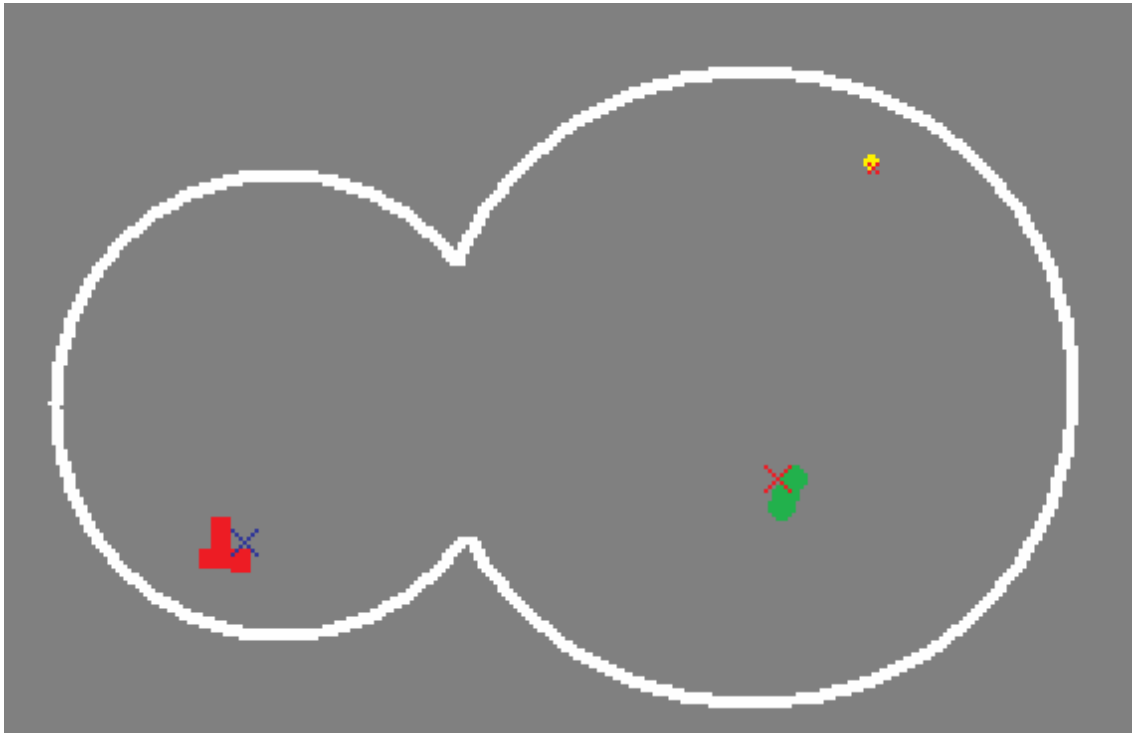


Figura 47: Resultados prueba 9

La imagen refleja las marcas encontradas y sus posiciones, representadas por una cruz azul en el caso de la marca roja (marca número 1), y por cruces rojas en el caso de la marca verde y la amarilla (marcas número 1 y 2). Por tanto, observando estos resultados, se puede concluir que la prueba 9 ha arrojado resultados satisfactorios.

En cuanto al rendimiento, al tratarse de un entorno más simple, no ha sido necesario modificar la trayectoria del robot en ningún momento para favorecer que encontrase las marcas. Sin embargo, el tiempo total que ha tardado en encontrar las 3 marcas ha sido de unos 7 minutos, debido a que ha encontrado varias veces las dos marcas más grandes (roja y verde), pero las ha obviado una vez que ha comprobado que la posición ya estaba almacenada. Esto ratifica la afirmación hecha en la prueba anterior de que la navegación es un aspecto a mejorar dentro del sistema global.

7. Conclusiones

En este apartado se expondrán las conclusiones generales obtenidas una vez realizado el proyecto y ejecutado todas las pruebas propuestas, a la vista de los resultados obtenidos.

Hay que destacar, antes que nada, el hecho de que el proyecto desarrollado ofrece una visión más original acerca del marcado topológico del terreno, ya que la mayoría de trabajos realizados sobre el tema tratan del marcado del terreno partiendo del conocimiento previo de las marcas que el robot está buscando; sin embargo, en este proyecto se ha intentado dar una visión totalmente automatizada del marcado del terreno, ya que el robot no posee ninguna información acerca del entorno por el que va a realizar la navegación. Este aspecto, aunque supone un reto interesante, también es la causa de que sea necesario acotar el alcance del proyecto para que éste sea razonablemente abarcable.

Relacionado con el aspecto anterior, en lo que se refiere al análisis de la imagen, el hecho de enfocar únicamente dicho análisis hacia operadores de color de la imagen se debe precisamente a esta acotación del alcance del proyecto. No obstante, el estudio realizado sobre el tema ha sido bastante completo y la técnica obtenida ha demostrado funcionar razonablemente bien sobre los entornos que se han utilizado para las pruebas. Lógicamente, se trata de un aspecto que puede fallar en muchas circunstancias y entornos, por lo que se definirán posibles mejoras en el apartado de trabajos futuros (Capítulo 8: Trabajos futuros). En general, mediante la comparación de diferentes técnicas se ha obtenido información interesante acerca de qué técnicas funcionan mejor que otras, además de valorar las diferentes ventajas e inconvenientes de utilizar cada una de ellas. Esta información se resume en la siguiente tabla:

Proyecto Fin de Carrera.
Sistema de atención visual para la detección de puntos topológicos de referencia.

TÉCNICA	VENTAJAS	INCONVENIENTES	RESULTADOS
Cruces por 128	Combina conceptos de análisis de señales de diferentes naturalezas (acústica, visual...).	Necesidad de equiparar la imagen a una señal continua. Mayor complejidad de implementación.	No satisfactorios
Histogramas	Técnica muy utilizada en análisis de imágenes.	Mayor complejidad de implementación. Necesidad de codificar el histograma.	No satisfactorios
Media de colores	Fácil de implementar. Intuitiva.	Necesita entornos muy simples para funcionar.	Satisfactorios en imágenes con colores homogéneos. No satisfactorios en imágenes con colores no homogéneos.
Colores más frecuentes	Intuitiva. Funciona relativamente bien en los entornos probados.	En entornos muy complejos o con muchos cambios de iluminación, da más problemas.	Satisfactorios en imágenes con colores homogéneos y no homogéneos

Tabla 20: Comparativa entre las técnicas utilizadas

En lo que se refiere al desarrollo del proyecto en sí, existe un aspecto que ha marcado fundamentalmente dicho desarrollo, que es el aprendizaje de la herramienta Microsoft Robotics Studio. Esta herramienta es una aplicación realmente potente y que ofrece una gran cantidad de posibilidades al desarrollador de robots, por lo que el

Proyecto Fin de Carrera.

Sistema de atención visual para la detección de puntos topológicos de referencia.

aprendizaje de los conceptos básicos que utiliza y de las ideas en las que se basa (CCR, DSS, *Runtime*, etc., todas ellas explicadas en el apartado de herramientas) requiere una gran cantidad de tiempo y trabajo. Sin embargo, el resultado final en este aspecto ha sido positivo, ya que aunque el desarrollo del proyecto ha llevado más tiempo del planificado inicialmente, los resultados obtenidos han sido satisfactorios, además de haberse conseguido un conocimiento bastante amplio acerca de la herramienta, lo cuál puede ser de gran utilidad en un futuro.

En general, se considera que se han cumplido los objetivos marcados al inicio del proyecto, tanto en lo que se refiere a funcionamiento de la aplicación como a los requisitos establecidos, ya que se ha construido una aplicación robusta, adaptable a diferentes entornos, fácilmente reutilizable y modificable, etc. Se han cumplido también los requisitos funcionales de la aplicación, como por ejemplo la realización del estudio comparativo entre las diferentes técnicas aplicadas, la definición de datos de entrada y de salida, el uso de navegación local y global, etc.

Como se ha explicado a lo largo del proyecto, la aplicación se divide en dos fases fundamentales: la fase de detección y la fase de localización. Como es lógico, los mayores esfuerzos se han centrado en la fase de detección, ya que es en la que se basa principalmente el proyecto; sin embargo, la fase de localización también ha supuesto un gran esfuerzo para conseguir resultados aceptables que presenten los mínimos fallos posibles.

8. Trabajos futuros

Existen multitud de ampliaciones y mejoras que se podrían realizar sobre el proyecto desarrollado. A continuación se expondrán las que se han considerado más importantes o interesantes de realizar:

En lo que se refiere a la fase de detección, existe una gran cantidad de técnicas orientadas al análisis de color en imágenes, además de las que se han utilizado en este proyecto, como pueden ser la utilización de diferentes filtros, disponibles en todas las librerías de análisis de imágenes que se pueden encontrar, tanto de libre distribución como privadas. Estas técnicas no utilizadas, por tanto, podrían ser la base de nuevas pruebas sobre el software desarrollado.

Existen también muchas más técnicas para el análisis de imágenes y detección de marcas que no se centran en el análisis de color, sino en la orientación, la morfología de los objetos, etc., y que están orientadas también hacia conceptos de atención visual, tal y como se ha expuesto en el capítulo 2: Estado del arte. Por ello consideramos que el desarrollo de alguna de dichas técnicas sería una buena opción, así como la combinación de técnicas de diferente naturaleza.

Dentro de la fase de localización, se podrían realizar mejoras muy importantes en los dos modos de navegación del robot, ya que teniendo en cuenta que el proyecto y la aplicación que se querían desarrollar no se centraban en ese aspecto, se han realizado algoritmos más sencillos para conseguir el objetivo esperado, pero sin centrarse específicamente en tiempos y recursos utilizados. Tal y como se ha explicado en el estado del arte, existen muchos algoritmos de navegación y localización de un robot (diagramas de Voronoi, grafos de visibilidad, modelado de espacio libre...) que se podrían utilizar para conseguir que la localización de las marcas fuese más rápida y eficiente. Sin embargo, como ya se ha dicho, no se ha considerado este aspecto como uno de los más importantes en lo que se refiere al desarrollo de este proyecto.

Relacionado con lo comentado anteriormente, puede ser muy interesante la inclusión de nuevos servicios en el control del robot que se complementen con el

Proyecto Fin de Carrera.

Sistema de atención visual para la detección de puntos topológicos de referencia.

servicio de marcado del terreno, como por ejemplo servicios que permitan detectar y evitar obstáculos en una navegación local, servicios que reduzcan la influencia de las variaciones de luz en el entorno analizado, etc.

Un trabajo futuro muy interesante consiste en la realización de pruebas del robot y de la aplicación desarrollada sobre entornos reales, para comprobar cuál es el nivel de acierto de las técnicas de análisis de imagen desarrolladas sobre dicho tipo de entornos. El paso de la simulación a entorno real no tiene por qué ser muy complicado si se dispone del hardware necesario, sin embargo por falta de tiempo no ha sido posible realizarlo en este proyecto. En el caso de que se realizasen pruebas del sistema en entornos reales, hay que tener en cuenta que el rendimiento sería peor que en entornos simulados, debido a la mayor complejidad de las imágenes que recibiría el robot. Esta complejidad viene dada por la mayor riqueza de colores, texturas y variaciones de luz existente en el mundo real respecto del mundo simulado utilizado en este proyecto.

En general, y como es lógico, cualquier mejora que se realizase sobre el proyecto y que significase un mayor rendimiento de la aplicación tanto en lo que se refiere a tiempos de ejecución como a recursos utilizados o a eficacia de la propia aplicación, sería interesante, siempre que no supusiera un trabajo demasiado complejo en comparación a los resultados que se obtuviesen.

También es interesante analizar cómo se podría integrar el servicio de marcado de terreno que se ha desarrollado, sobre otro servicio de control que no fuese el que se está utilizando en este proyecto. Teniendo en cuenta que se intenta mantener la estructura del servicio de marcado de terreno, los datos de salida que éste proporciona (la imagen original y la matriz binaria con las regiones marcadas) deberían respetarse. Por tanto, el servicio de control que hiciera uso de este servicio de marcado tendría que utilizar estos datos para realizar las funciones deseadas. Se puede pensar, por ejemplo, en un servicio de control que no necesitase acercarse a la marca para almacenar su imagen, sino que, para eliminar el tiempo empleado en dicha aproximación, tuviese implementados métodos más eficaces para medir la distancia hasta la marca desde cualquier posición, solventando posibles errores que se pudiesen cometer.

Glosario de términos

- API: Application Programming Interface.
- ARIA: Interfaz Avanzado de Aplicaciones Robóticas.
- ASP: Active Server Pages.
- CCD: Charge-Coupled Device.
- CCR: Concurrency and Coordination Runtime.
- CLR: Common Language Runtime.
- CMOS: Complementary Metal Oxide Semiconductor.
- CPU: Control Processing Unit.
- CRG: Cilindros Rectilíneos Generalizados.
- DLL: Dynamic Linking Library.
- DSS: Decentralized Software Services.
- DSSP: Decentralized Software Services Protocol.
- GLOH: Gradient Location and Orientation Histogram.
- GPS: Global Positioning System.
- HSL: Hue – Saturation – Luminance.
- HTTP: HyperText Transfer Protocol.
- IA: Inteligencia Artificial.
- LRF: Laser Range Finder.
- MSRDS: Microsoft Robotics Developer Studio.
- PTZ: Pan Tilt Zoom.
- RGB: Red – Green – Blue.
- ROI: Region of Interest.
- SIFT: Scale-invariant feature transform.
- SO: Sistema Operativo.
- SONAR: Sound Navigation And Ranging.
- STM: Short-Term Memory.
- TCP/IP: Transfer Control Protocol / Internet Protocol.
- VB: Visual Basic.
- VPL: Visual Programming Language.
- XML: Extensible Markup Language.

Bibliografía

- Mata, M. et al.: “A visual landmark recognition system for topological navigation of mobile robots”, 2001.
- Bachiller, P.: “Percepción dinámica del entorno en un robot móvil”, 2008.
- Diccionario de la Real Academia Española, 2009.
- Brooks, R.: “Elephants don’t play chess”, 1991.
- Siegwart, R.: “Introduction to autonomous mobile robots”, 2004.
- Bernabé, I.: “Sistema de detección visual de obstáculos para un robot móvil autónomo”, 2007.
- Nilsson, N. J.: “Shakey the robot”, Menlo Park, 1969.
- Herrero, I.: “Control de blisters mediante visión artificial”, Universidad Nacional de Quilmes, 2005.
- James, W.: “The principles of psychology”, 1890.
- Perry, R. y Hodges, J.: “Attention and execution deficits in aliheimer’s disease: A critical review”, 1999.
- Broadbent, D. E.: “Perception and communication”, 1958.
- Deutsch, J. y Deutsch, D.: “Attention: Some theoretical considerations”, 1964.
- Allport, A.: “Selection for action: Some behavioural and neuropsychological considerations of attention and action”, 1987.
- Riddoch, M. et al.: “Visual affordances and object selection”, 2000.
- Treisman, A. M. y Gelade, G.: “A feature-integration theory of attention”, 1980.
- Wolfe, J. M.: “Guided search 2.0: A revised model of visual search”, 1994.
- Cave, K.: “The featuregate model of visual selection”, 1999.
- Posner, M. y Delaene, S.: “Attentional networks”, 1994.
- Koch, C. y Ullman, S.: “Shifts in selective visual attention: towards the underlying neural circuitry”, 1985.
- Itti, L. y Koch, C.: “A saliency-based search mechanism for overt and covert shifts of visual attention”, 2000.
- Tsotsos, J. K.: “Modelling visual attention via selective tuning”, 1995.
- Sun, Y. y Fisher, R.: “Object-based visual attention for computer vision”, 2003.

Proyecto Fin de Carrera.

Sistema de atención visual para la detección de puntos topológicos de referencia.

- Frintrop, S.: “Goal-directed search with a top-down modulated computational attention system”, 2005.
- Hayet, J.B. et al.: “A visual landmark framework for indoor mobile robot navigation”, 2002.
- Frintrop, S. y Jensfelt, P.: “Attentional landmarks and active gaze control for visual SLAM”, 2008.
- Dodds, Z. y Hager, G. D.: “A color interest operator for landmark-based navigation”, 1997.